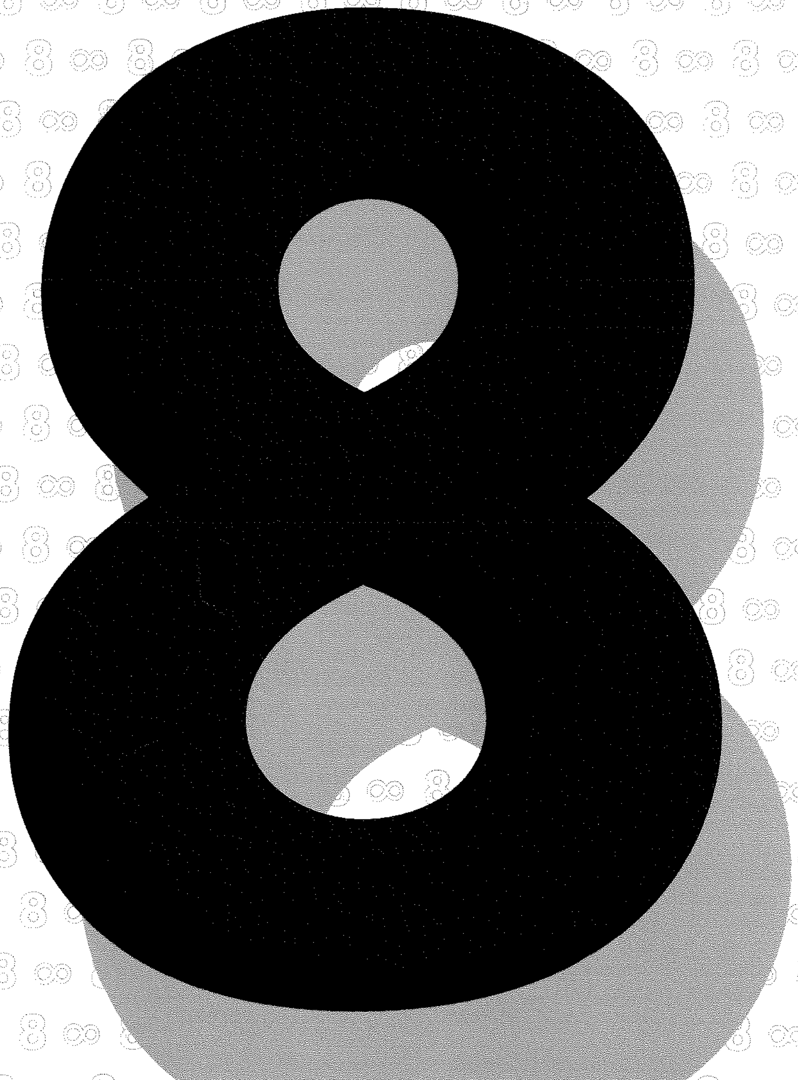
QL Today

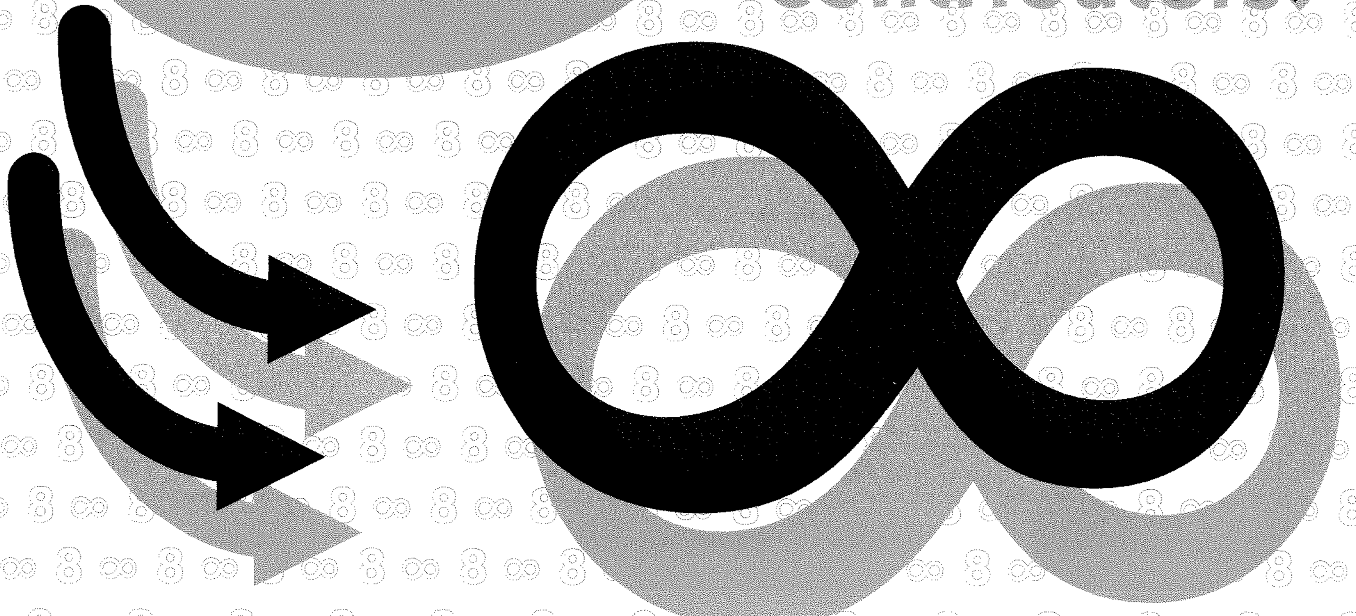
Volume 8
Issue 2
May/June
2003

ISSN 1432-5454

The Magazine about QL, QDOS,
Sinclair Computers, SMSQ...



It seems not too unlikely – with your help QL Today has moved into its 8th year – thanks to you and our contributors!



Contents

Advertisers

in alphabetical order

QL *Today*

ISSN 1432-5454

German office & Publisher:

Jochen Merz Software
Im stillen Winkel 12
47169 Duisburg
Germany

Tel. +49 203 502011
Fax +49 203 502012
BBS +49 203 502014
email: JMerz@j-m-s.com
email: QLToday@j-m-s.com

English office:

Q Branch
20 Locks Hill
Portslade
BN41 2LB
United Kingdom

Tel. +44 1273 386030
Mobile +44 7836 745501
Fax +44 1273 381577
email: qbranch@qbranch.demon.co.uk
email: QLToday@j-m-s.com

Editor:

Dilwyn Jones
41 Bro Emrys
Tal-Y-Bont, Bangor
Gwynedd
United Kingdom LL57 3YT

Tel. +44 1248 354023
email: dilwyn.jones@tesco.net
email: QLToday@J-M-S.com

Co-Editor:

Bruce Nicholls
38 Derham Gardens
Upminster
Essex RM14 3HA
United Kingdom

Tel +44 1708 510764
Fax +44 870 0568755
email: qltoday@q-v-d.demon.co.uk
email: QLToday@j-m-s.com

QL *Today* is published bi-monthly, our volume begins on beginning of June. Subscriptions begin with the current issue at the time of sign up. Please contact the German or English office for current subscription rates.

We welcome your comments, suggestions and articles. YOU make **QL *Today*** possible. We are constantly changing and adjusting to meet your needs and requirements. Articles for publication should be on a 3.5" disk (DD or HD) or sent via Email or into one of the JMS-BBS's. We prefer ASCII, Quill or text87 format. Pictures may be in _SCR format, we can also handle GIF or TIF or JPG. To enhance your article you may wish to include Saved Screen dumps. PLEASE send a hardcopy of all screens to be included. Don't forget to specify where in the text you would like the screen placed.

Article and Advertising deadlines are as follows:
Issue 1: 30 April
Issue 2: 30 June
Issue 3: 30 August
Issue 4: 30 October
Issue 5: 30 December
Issue 6: 28 February

QL *Today* reserves the right to publish or not publish any material submitted. Under no circumstances will **QL *Today*** be held liable for any direct, indirect or consequential damage or loss arising out of the use and/or inability to use any of the material published in **QL*Today***. The opinions expressed herein are those of the authors and are not necessarily those of the publisher.

Editorial

Dilwyn Jones

With this issue we begin our eighth year of publishing QL Today, which is no mean achievement considering that the original QL was last manufactured nearly twenty years ago!

This issue may arrive a bit late. As that is not the normal state of affairs for QL Today, please read Jochen's message on page 3. You'll see he's been ill of late and true to form has not made a public fuss of it at all up to now. That's the Jochen I've known all these years, originally as a customer of his in the early days of the QL then as editor and friend in more recent years. I'd like to use this column to wish him a speedy recovery and a vote of thanks for all the hard work he has tirelessly put in for the QL over the years. We are very lucky to have Jochen Merz right here as a driving force at the heart of the QL scene.

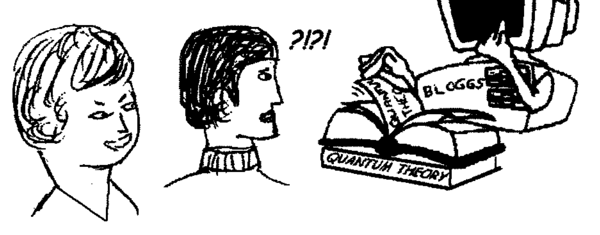
The latest QL emulator recently hit the QL World in the form of QL2K, a port by Jimmy Montesinos of the original QLay emulator by Jan Venema. QL2K is targetted at WIndows 2000 and Windows XP users, but seems to work on Windows 98 and Windows Me as well. Although released as Postcardware, it's basically free, all you have to do is send your name and address to the author to register yourself as a user!

An unashamed mention here for Jim Hunkins and myself. We have both been beavering away on separate Graphical User Interfaces for the QL. Well, it's high time the QL got one (or in this case, two). Jim's QDT system is an unashamedly high end SMSQ/E and GD2 project, whereas my system (Launchpad) is an altogether less ambitious project, more of a compromise for both QDOS and SMSQ/E systems. Both projects look like coming to fruition at similar times, and I hope to have more detailed news on both systems in the very near future for you.

In the meantime, I've used experience I've gained from writing Launchpad to produce an introductory article on using the new "colour drivers" from SBASIC (or GD2 to use the official name) in this issue, for those readers fortunate enough to be using suitably equipped systems like Qx0, QPC2 and QXL.

And what with SMSQ/E v3 and GD2 for Aurora with 256 colour support both due out very soon, and Jonathan Dent dropping occasional little progress reports on his latest achievements with his SOQL TCP/IP system, it'll be an exciting few months ahead for QLers I'm sure!

Cartoon



I knew that these QL systems are getting smart, but this is ridiculous!!!

Thank you!

Jochen Merz

Glad to see you're all there. The major renewal after last issue showed that the large majority of our readers stayed with us. We lost very few, but also gained some new readers. This is very encouraging and we take it as an expression from you, our readers, that you like what we are doing.

The current issue is rather late. I know. I have informed every re-subscriber about this fact. Two months ago I was still hoping to have this issue ready early, maybe even for the US show. The material situation however, did not make this possible at the time - most articles arrived late. Nevertheless, we have a full, good looking issue - as we do most of the time thanks go to our trusty authors. Special thanks also go to Dilwyn, Bruce and Roy for their permanent support and help - without all of you QL Today wouldn't be possible!

But this is not the only reason for this issue being late - I had to go into hospital (again) fairly unexpectedly, and this and the related problems, visiting more doctors and so on have taken more time than I was ever expecting.

I am trying to avoid another operation under every circumstance, so I have to slow down. This not only tells me my body, the doctors say so too. In the past, even when things were late, the responsiveness was poor etc. I put in every possible effort to have the magazine out in time, often even earlier to make it available at a show. I will continue to ***try*** to have it ready in time but I will give up on putting other problems, no responses until last-minute etc. onto my back.
But don't get the wrong impression: I'm not blaming anybody, sometimes the problems just add up because everybody thinks it's only him who is late.

You surely know by now (after 7 years of QL Today and many more years in the QL scene) that we're giving our best efforts to get the magazine out in time - if it is a bit later in the future, every now and then, then it is not our fault unless we state it is.

We're always trying very hard to compile an interesting magazine, a good mix and as many pages as possible, but remember - we can only print material we get...

Finally I would personally like to say a Thank You very much to the QLers who sent me a "Get well soon" card or letter after they received my information of the possible delay of this issue. Not just a phrase - I was very touched by it and enjoyed receiving them very much, although the reason itself is not very enjoyable.

Whatever happens - there will be more issues of QL Today, even if one or the other may be a bit later than expected ... but: health comes first, I've learned my lesson and I am sure you understand.

I am also still coming to every possible QL show, and although the earnings drop and the travelling costs increase, the work and stress involved remains the same (well, stress is the wrong word, every show is enjoyable, but there is still the preparation, the travelling and so on). There are more shows planned in autumn and I'll be at every show too (Berchtesgaden, Eindhoven, Italy and Byfleet), but as I need to recover in the time between and not work 16 hours a day then, things may be delayed a bit too.
Don't get me wrong here: I am not trying to be lazy, but if the workload is just too high, then things may have to wait.

And talking about waiting: yes, I was hoping to have all the new programs out months ago. They were more or less ready by the end of February, after Bernd and I went skiing, and it was a bit disappointing not having had them ready at the Hove or USA show, but we had to wait for SMSQ/E Version 3 to be completed and released. Marcel and Wolfgang were still tweaking bits here and there and the other programs had to be slightly adjusted here and there to accomodate for the changes too. No lazyness here, just trying to give away final, properly working products which need no complete replacement a few days later.

All the changes are part of a large opportunity to get many new things sorted out now and they were also a path for future definitions, so careful decisions had to be made what to do how, and to make sure they were also compatible with existing software. Not an easy task, as many of you know, but the result is worth the effort and the wait.

I for myself am looking positive forward to another QL (Today) year!

NEWS

Just Words!

has recently extended its range of word lists and QTYP dictionaries.

New products are:

VOCABULARY DATABASE: This is a database of over 5,000 words in English, German, French, Dutch and Japanese. The database has been compiled from several language courses to give a vocabulary equivalent to about A Level knowledge.

The database is in both Archive _dbf and export format, so that it can also be imported into other database programs. This includes, after conversion by QL-2-PC Transfer, PC database programs.

As it is easy to add or remove fields the database may also be interesting to students of other languages, who could add new words as their knowledge of the language increases.

VOCABULARY DATABASE costs £5 or 7.50 Euros.

SCRABBLE LISTS: These are two lists of Scrabble permitted words. OPSD has 79,339 words of maximum length 8 letters. This list is for the player of average ability. ENABLE is a larger list of 173,252 words of maximum length 19 letters. This list is more suitable for advanced players and Scrabble competitions. Both lists can be easily imported into Solvit-Plus.

SCRABBLE LISTS costs £1 or 1.50 Euros.

QTYP DICTIONARIES: There are four new QTYP dictionaries. These are an ENGLISH dictionary of 194,433 words (mixed UK and USA spellings); a DUTCH dictionary of 180,130 words (conforms to 1995 spelling revision); an ITALIAN dictionary of 83,829 words; and a GERMAN dictionary of 165,935 words (old spellings).

Each dictionary costs £1 or 1.50 Euros.

Just Words! hopes to release more QTYP dictionaries and word lists in the near future.

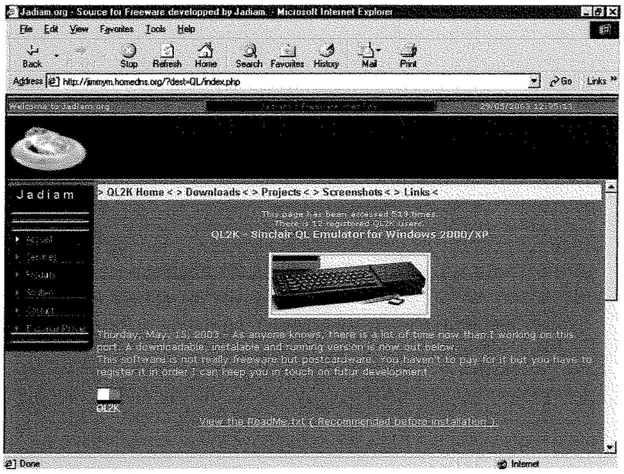
QL2K

Developed by ***Jimmy Montesinos*** from the sources of ***Jan Venema***'s QLay emulator, QL2K is an enhanced QLay for Windows 2000 and Windows XP systems. It also seems to run on Windows 98 SE and Windows Me, although at the time of writing the latter had not been tested. Like QLay, QL2K is basically free, but is offered as postcardware. In other words, it is free but Jimmy Montesinos asks that you send him an email or postcard with your name and address to register your copy (free) so that he can keep you informed of future developments.

QL2K adds long filename support for WIN drives and a release of an English menu for DirectDraw and hot configuration support. Full screen will be supported via Direct-X support and up to 8 WIN and emulated MDV drives available. Jimmy is releasing information via the ql-users email mailing list on progress as he is actively developing the alpha releases as I write this item. We also hope to receive an article from him about the emulator. At the time of writing, release of version 0.94 alpha was imminent.

Jimmy has built a website dedicated to QL2K, which may be found at:

http://www.jadiam.org/QL/index.php



Jadiam.org - Source for Freeware developped by Jadiam. - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Stop Refresh Home Search Favorites History Mail Print
Address http://jimmym.homedns.org/?dest=QL/index.php
Go Links
Jadiam
> QL2K Home < > Downloads < > Projects < > Screenshots < > Links <
QL2K
Done
Internet

This page has been accessed 513 times.
There is 12 registered QL2K users.
QL2K - Sinclair QL Emulator for Windows 2000/XP

Thursday, May, 15, 2003 - As anyone knows, there is a lot of time now than I working on this port. A downloadable, instalable and running version is now out below.
This software is not really freeware but postcardware. You haven't to pay for it but you have to register it in order I can keep you in touch on futur development.

View the ReadMe.txt (Recommended before installation)

Jimmy's QL2K website

QL Emulators CD

Release 1.23 of the QL Emulators CD from ***Dilwyn Jones*** includes the new QL2K emulator from Jimmy Montesinos and should be available in early June.

Keyboard Membranes

Rich Mellor has announced that the new production run of QL keyboard membranes is now on the way and will be available very soon at a price of £18.50 each plus postage from RWAP Software. At the time of writing Rich Mellor had told us that they were on their way from Germany and samples tested had proved to be of excellent quality.

SMSQ/E V3 Sources

SMSQ/E v3.00 is now available, writes ***Wolfgang Lenerz***. You can get the upgrade from the usual sources. Speaking of sources, the sources for smsqe 3.00 will be available very shortly, just send the cd back to Wolfgang (he is the SMSQ/E sources registrar).
If you want to get an idea on how good things look under SMSQ/E V3, go to the JMS ad and check out the screendumps of QD and QSpread.

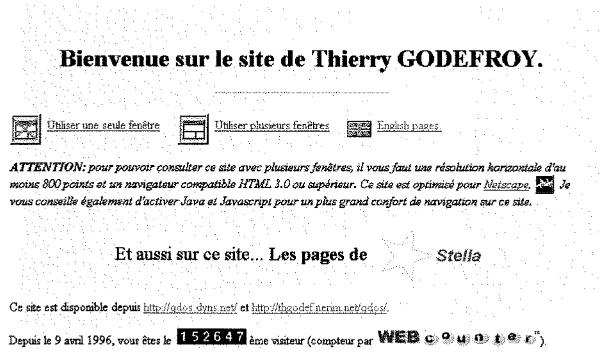
HELP

Georgio Garabello writes about his HELP utility: Help 1.2 is now available, fixed some minor bugs + documentation updated. Download from:
http://utenti.lycos.it/Sinclair/HELP12.ZIP

Thierry Godefroy

After nearly one and a half years away from the QL scene mainly due to work commitments, ***Thierry Godefroy*** is back. His website is partly updated (the new software repository is up too). You'll find everything on the new addresses below (please update your bookmarks):

http://qdos.dyns.net/ (main site)
http://thgodef.nerim.net/qdos/ (mirror of main site)
http://thgodef.nerim.net/smsq/ (software repository)



Bienvenue sur le site de Thierry GODEFROY.

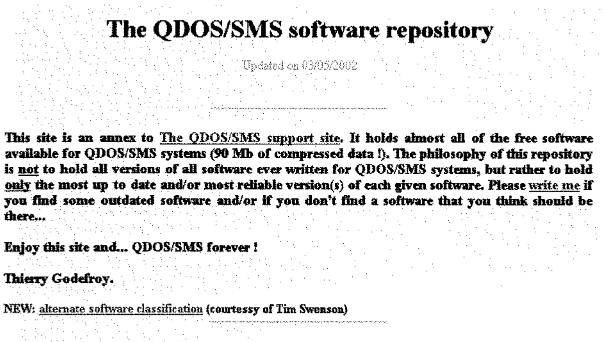
ATTENTION: *pour pouvoir consulter ce site avec plusieurs fenêtres, il vous faut une résolution horizontale d'au moins 800 points et un navigateur compatible HTML 3.0 ou supérieur. Ce site est optimisé pour Netscape. Je vous conseille également d'activer Java et Javascript pour un plus grand confort de navigation sur ce site.*

Et aussi sur ce site... **Les pages de** *Stella*

Ce site est disponible depuis http://qdos.dyns.net/ et http://thgodef.nerim.net/qdos/.

Depuis le 9 avril 1996, vous êtes le 152647 ème visiteur (compteur par WEB counter).

Thierry Godefroy's website



The QDOS/SMS software repository

Updated on 03/05/2002

This site is an annex to The QDOS/SMS support site. It holds almost all of the free software available for QDOS/SMS systems (90 Mb of compressed data !). The philosophy of this repository is not to hold all versions of all software ever written for QDOS/SMS systems, but rather to hold only the most up to date and/or most reliable version(s) of each given software. Please write me if you find some outdated software and/or if you don't find a software that you think should be there...

Enjoy this site and... QDOS/SMS forever !

Thierry Godefroy.

NEW: alternate software classification **(courtessy of Tim Swenson)**

The menu:

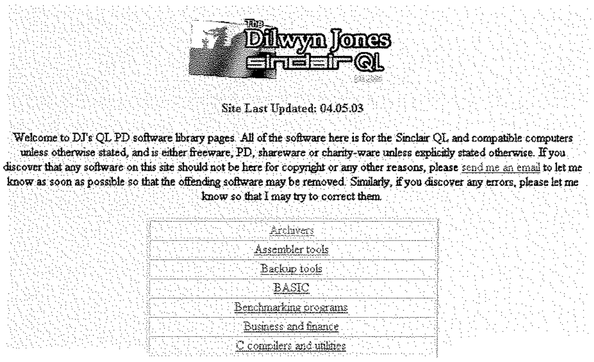
Thierry Godefroy's Software Repository

LPSaver

I have released the LPsaver screen saver utility from Launchpad as a freeware stand alone screen saver utility. This is a mouse and keyboard aware program which requires pointer environment and Toolkit 2. The program comes with a dozen or so built in screen saver modules ranging from a straightforward screen blanker to a colourful GD2-aware routine called Dayglow. Most of these modules will work on anything from an expanded memory QL with pointer environment up to QPC2 and Q40/Q60. You can also write your own modules - just about anything executable and reasonably well behaved should work. SBASIC users can even write modules as a BASIC program which is executed by LPsaver. Once the screen saver has cut in after a configurable time of inactivity, moving the mouse, pressing a mouse button or a key on the keyboard will stop it again and restore the original display.
When Launchpad becomes available, the screen saver will be controllable from the My-QL menu, or LPsaver can be used standalone as required (you do not have to be running Launchpad to use LPsaver).
LPsaver can be downloaded from my website on:
http://homepages.tesco.net/dilwyn.jones/software/freeware/freeware.html

QL Software Download Site

With greatly appreciated assistance from ***Phoebus R. Dokos***, I have moved what used to be known as the Other Software Page from my old website to a new and much enhanced presence on **http://www.dokos-gr.net/~dj/index.html**
There are links from my other 'Tesco.net' site to this new software download site.
At the time of writing the site is incomplete but a substantial part of it is already there and working. Over time, I hope this will become a significant source of free QL software and to put as much of my PD library as possible onto this site for free download.



Site Last Updated: 04.05.03

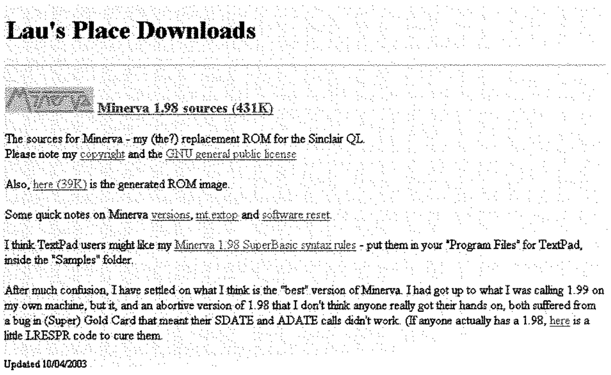
Welcome to DJ's QL PD software library pages. All of the software here is for the Sinclair QL and compatible computers unless otherwise stated, and is either freeware, PD, shareware or charity-ware unless explicitly stated otherwise. If you discover that any software on this site should not be here for copyright or any other reasons, please send me an email to let me know as soon as possible so that the offending software may be removed. Similarly, if you discover any errors, please let me know so that I may try to correct them.

Archivers
Assembler tools
Backup tools
BASIC
Benchmarking programs
Business and finance
C compilers and utilities

Minerva

Laurence Reeves has now made the sources and binary for Minerva version 1.98 available via his website on **http://www.bergbland.info**

Minerva bites back?



Lau's Place Downloads

Minerva 1.98 sources (431K)

The sources for Minerva - my (the?) replacement ROM for the Sinclair QL.
Please note my copyright and the GNU general public license

Also, here (39K) is the generated ROM image.

Some quick notes on Minerva versions, mt.extop and software reset.

I think TextPad users might like my Minerva 1.98 SuperBasic syntax rules - put them in your "Program Files" for TextPad, inside the "Samples" folder.

After much confusion, I have settled on what I think is the "best" version of Minerva. I had got up to what I was calling 1.99 on my own machine, but it, and an abortive version of 1.98 that I don't think anyone really got their hands on, both suffered from a bug in (Super) Gold Card that meant their SDATE and ADATE calls didn't work. (If anyone actually has a 1.98, here is a little LRESPR code to cure them.

Updated 10/04/2003

Laurence Reeves Minerva web page

Turbofix

Davide Sanatachiara has released a small update to the Turbofix extensions that are supplied with the Masterbasic Superbasic development utility. Turbofix is a collection of Turbo Toolkit keywords (BASIC_*) that have been amended in order to work with Minerva Multibasics and SMSQ/E SBasics (work done many years ago together with Rich Mellor). This fix allows multiple copies of MasterBasic to run on multiple SBasics for instance. Unfortunately due to some incompatibilities Turbo did not work anymore after loading Turbofix extensions. This has now been corrected thanks to George Gwilt who found the problem. The latest Masterbasic package available on Davide's web site:

www.geocities.com/dsantachiara

contains Masterbasic and the updated Turbofix release (3.0d). Thanks to Turbofix it is also possible to compile Turbo programs from SBasics (just load Turbofix and then use EW parser_task:EW codegen_task - parser and codegen must be owned by the SBasic interpreter). Parser_task and Codegen_task run fine and compile correctly SBasic programs but it seems there is still a bit of further investigations as Davide had some problems compiling MasterBasic itself from SBasic. Davide also reminds us that the DEA intelligent disassembler has been updated for GD2 compatibility.

SINTECH

Jean-Yves Rouffiac writes:

Just came across this site which has a stack of Spectrum/QL/Z88/ZX81 and non-Sinclair stuff for sale, both h/w and s/w. I don't think that it's on QLNet, though I may have missed it under a different title.

They have QL software for sale from the like of Eidersoft! Ah, memories...

http://www.sintech-shop.de/home/default.html

Small Ads

QL Parts for sale

4 original cased QL's with MGD-rom and other parts:

1x Super Gold Card and Hermes (4 MB), 1x Gold Card and Hermes (2 MB), 1x Trumph Card (0,86 MB), 1x 512 K Expanderram (0,640 MB) with crashed screen ula. All with power units.

2x Replaced 8049IPC relayed. 2x Spare QL keyboard membrans. 1x English and 2x Danish user manuals. 1x Miracle 40 MB harddisk newformatted with startup PE-system. 1x Miracle 40 MB harddisk with crashed harddisk else OK. 1x Miracle with 2 DD floppy drives with built in power unit. 2x Sinclair DD floppy drives with seperate power units. 2x Micro Pheripherals interfaces for floppy drives. 1x Euro Modem 1200 with power unit. 2x QLink to matrix printers. 3x CARE TK2 ROM. 3x 3 button Mouses. 4x NET, 5x Mono screen, 2x colour screen wirings. 6x Ser & Ctrl wirings. Ca. 250 DD floppys formatted or program floppys. Ca. 150 MDV formatted or program MDV's. 2x Keymax monitor consols. 1x 15" Diaspron screen filter.

Someone out there interested in all or some of my good old QL stuff, better to be used than laying collecting dust. I have not left the QL-world but are using QPC2 on PC and laptop. QL forever!

Joergen Rasbo, Bausager 6, Gevninge, DK 4000 Roskilde, Denmark.
Mail: **jrasbo@tdcadsl.dk** Phone: +45 4640 2233

Gee Graphics! (on the QL ?) - Part 33

H. L. Schaaf

"Steiner or Gauss? or which way to run a railroad?"

The minimum spanning tree (MST) was mentioned before in GG#30

By adding more "Steiner" points, it is possible to have a shorter MST and thus 'more makes less'. Fermat (1601-1665) posed this problem: Given 3 points of a triangle, find a 4th point such that the sum of its distances to the 3 points is a minimum. Fermat's problem is easy to ask, but can prove difficult to answer. Before 1640 Torricelli solved "Fermat's problem", and the added point came to be known as "Torricelli's point".

In the 19th century it was expanded to "Steiner's problem": given any number of points in the plane, find the "Torricelli" point where the sum of the distances to each of the old points is a minimum. A problem with only 4 points was discussed in correspondence between Gauss and Schuhmacher in 1836. A railroad was to be built connecting the 4 cities of Harburg, Bremen, Hannover, and Braunschweig. Which way should the railroad go to minimize the length? This could (should?) be called "Gauss' problem".

Gauss showed that the Torricelli point for Harburg, Bremen and Hannover should be a common junction point for those 3 cities, and then the line from Hannover to Braunschweig added.

How might we look at this on the QL?

Take the meridian of longitude 10 degrees East as a 'vertical' reference and the parallel of latitude 53 degrees North as a 'horizontal' reference. Scale in kilometers to come up with the following coordinates:

Place	"X"	"Y"
Harburg	0	+50
Bremen	-81	+10
Hannover	-17	-67
Braunschweig	+36	-81

Use the Keyboard input option of GG#30 or GG#31 to enter these 4 points, see what the MST looks like and note the length.
The Torricelli point for Harburg, Bremen, Hannover is -41, +2.4 and a nearby city is Rotenburg/Wumme, located at -40, +12.

Repeat for 5 points, with Rotenburg/Wumme as the fifth point and note the length of the MST.

Is it worth the difference? What's the cost of building and maintaining a dozen kilometers of track, telegraph, etc.?

Naturally in real life the railroad location engineer has to take many other factors into account such as rivers, marshes, mountains, politics, etc. Gee-ography, Gee-ology, Gee-Graphics.

I'd be curious to know if the railroad was built, and if so, where did it actually go? Anybody know?

Next time - how to use the QL to find the "Torricelli" point for 3, 4, and maybe more points, and why "Gauss' problem" became known as the "Steiner tree problem" and how Torricelli points became known as Steiner points.

References:

1 - "The Steiner Tree Problem" FK Hwang, DS Richards, P Winter ISBN: 0 444 89098 X Elsevier Science Publishers B.V. 1992

2 - "Steiner Minimal Trees" Dietmar Cieslik ISBN: 0-7923-4983-0 Kluwer Academic Publishers 1998

Transferring Spreadsheets

Geoff Wicks

If you are a gruesome and sadistic sort of person, continue reading. You are about to witness a QL trader committing commercial hara-kiri.
The ceremonial sword is now ready. So here it comes:

"If you want to transfer QL files to a PC, you do not have to buy QL-2-PC Transfer."

There. The deed is done. My QL soul has transcended to a higher level of virtual reality and now resides in a recursive loop of eternal peace.

Recently I received a telephone call from a freelancer, who for years has done his accounts in Abacus, but whose accountant now wants them in Excel format. He asked if he could transfer them, and was surprised to hear that as a

QPC user he already had the software to do the job. However he would only be able to transfer the values and not the formulae. He would also be unable to transfer accented letters correctly without additional software.

When I explained how to do this I was speaking from memory and gave him some inaccurate information about the PC side of the transfer. When he phoned again, I had to re-check the procdure. As I know other people who are interested in transferring spreadsheets I thought it worthwhile to write it up for QL Today. I shall also add a few comments about transferring the text from word processing files.

All the Psion programs have an Export command that allows you to transfer information between them, although with Quill it is present in the Xchange version only. You can also use this export command to transfer files from Abacus to a PC spreadsheet. Load your spreadsheet into Abacus then press F3, F for Files and E for Export. By default the export is to Quill, although you can press A to export to either Archive or Abacus or E to Easel. In our case we need to export in Quill format and thus we just press enter. You are then asked to enter the range of the spreadsheet you wish to export, and then the file name for the exported file.

Abacus will normally save the exported file to your default drive and add the extension _exp, but you can override this. For example you could enter ram1_myfile_txt. Ram1_ is a good device to choose as eventually you must save the exported file to a PC formatted disk, and Abacus will not allow you to do this directly. You should also override the file extension _exp with one that your PC spreadsheet uses to import files. This is an important point because this was where I mislead the freelancer on the telephone. Now save the file from ram1_ to a PC formatted floppy and don't forget to change the underscore to a full stop:

e.g.

```
COPY ram1_myfile_txt TO
flp1_myfile.txt
```

[You cannot do this without running SMSQ/E, by the way - Editor]

If you are just wanting to incorporate your spreadsheet in a document, you could simply import this text file into your PC wordprocessor. To import into a PC spreadsheet, you will need to know how your PC spreadsheet recognises when a file must be imported and not simply loaded. I do not possess Excel, but Lotus 123 does this when the file extensions are .TXT, .PRN, .CSV, .DAT, .OUT and .ASC. Note that .EXP is not on this list and that is why you cannot use this extension when exporting from Abacus.

When you load your export file into Lotus123 you are asked what separator you are using to start a new column. It suggests TAB, which is what we want, and when we confirm this our QL spreadsheet appears on the PC.

Now for a little complication. Many people use spreadsheets for financial calculations and if you try to transfer these you will find your pound signs have been replaced by apostrophes. This is not a serious problem. Before you export from Abacus press F3 D for Design and M for Monetary Symbol. Now enter Ctrl+Shift+c and you will get chr$(163), which is a capital E with an acute accent. Press enter to confirm this and then export your file. If you work in Euros use CTRL + ESC to get chr$(128), a lower case a with an umlaut. If you work in dollars you should not need to make any changes.

There are various ways in which it is possible to transfer databases and spreadsheets, presumably including formulae, between PC programs, although I have never used these and know nothing about them. This is not possible from Abacus, but theoretically would it be possible to do this from the QL? Has anyone done it? QL Today is permanently in need of articles and it would be interesting to hear about the experiences of readers who regularly transfer databases and spreadsheets. In the history of QL Today there have been few articles on this subject and a revue of the QL alternatives to Abacus and Archive would also be interesting.

Finally I promised to tell you how to transfer word processing files from QL word processors to the PC without using QL-2-PC Transfer. Remember, however, that if you do this, you can only transfer text and not formatting, and that if you use accented characters these will not be translated to their Windows equivalent.

Let's start with the easiest, which is Text87. I am typing this article in Text87 and after I have spellchecked, style checked and finished it, I shall save it to a PC disk and send it by email to QL Today. All I have to do is put a PC formatted disk in flp1_, then press F3 F for Files and E for Export. I am then asked if I want to save as a Text87 or an ASCII file and I type A for ASCII. Then if I want to export tabs as tabs or spaces and I press T for Tabs. All I now have to do is to enter the drive and file name, which in this case will be "flp1_harakiri.txt"

Next in order of simplicity is Perfection. This has an export command accessed by F3 I for export flle. Be wary of this, because Perfection has some unusual formatting quirks and often replaces line feeds by Chr$(206). Also in order to get continuous ASCII text you need as few line feeds as possible. Thus always follow the following procedure before using the export command. Press F3 F3 W to turn line Wrap off. When you do this you will see an "NW" note at the bottom of the screen. Now press F3 F3 M to adjust the Margins, and set Left and Indent margins to 1 and the Right margin to 1024. Finally go to the top of the page (CTRL + T) and press F3 F3 E for reformat hEreon. Provided your paragraphs are not too long, each paragraph will now be a single line and you can safely use the export command.

Finally Quill is the most difficult. The Xchange version of Quill does have an export command, but I would strongly advise you not to use it. It is inflexible in its generation of line feeds and this can be a major headache if you want to reformat the text in a PC word processor. The best way of extracting text from Quill is Norman Dunbar's Stripper program, which comes in both a QL and a PC version. You can download this from Dilwyn Jones' website. No doubt Dilwyn will append his web address to this article, because as a matter of principle I could not possibly guide you to a rival product. OK, I confess. what I have just written is not true. I am just too lazy to look it up.

[We too (this time) - Editor]

Quanta AGM

Tony Firshman

Despite the trader's emailshot accidentally including a map for Victoria Road, PORTSMOUTH, the Quanta AGM and workshop at Victoria Road, Portslade was very well attended, lively, and busy.

Be ***very*** suspicious of **www.map24.co.uk** I entered 'Victoria Road, Portslade', its URL included 'city0=portslade', but the untitled map was of Portsmouth!

www.streetmap.co.uk is a lot better.

Bill Richardson and I shared my car, and despite a failed battle at Little Chef to get a satisfactory breakfast without beans (reminds one of spam), we arrived in good spirits. Yes - I agree. "Why on earth go to Little Chef?" Well you try finding better on a Sunday at 8am! It has never been the same since Rocco was ousted - I liked it when it was down market with faded dirty paper menus.

Bill had fallen asleep. He must have still been asleep when we were driving over the hill into Portslade and I announced "I see the sea". His first words were "Is this Croydon?". (sorry Bill).

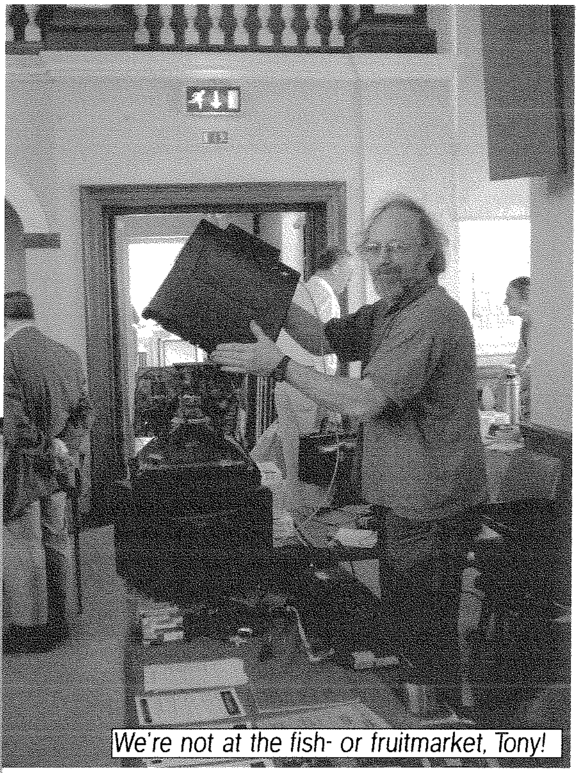
Portslade Town Hall is really very good - large, clean, enough power sockets, a vast kitchen, four attached committee rooms, and a stone's throw from Roy's house. No wonder he said "I wish I had found this years ago". It even has its own working pipe organ. Pity

QL Laptops everywhere

Busy QL show...

Stuart Honeyball was not there - he was an organ scholar.

The only real problem was the dazzling sun - "Why are you spending all day inside on such a day?" said Roy's boss, who's partner was assisting Roy's wife Saskia on the excellent food and drink stand. Julia (now 8) was as usual doing a very good sales job, and was now in charge of the money (clever girl). In fact the room seemed full of Roy's offspring, including at least one grandchild!


We're not at the fish- or fruitmarket, Tony!

Making sure it's still working? It is!

I found I had forgotten all my I2C boxes. Roy had a set he was given in the USA last year. I used those to get Ben's Lego working, and promptly sold all Roy's borrowings within 30 minutes of the show starting! (thanks and sorry Roy - I am manufacturing again).

I even sold a Microvitek colour monitor to someone who has lived for 20 years with a recently expired green monitor. I am sure he felt the struggle home by train was worth it. I can still remember my move from green mono to modified Ferguson TV. That was a relief for me after only a few years, but TWENTY years in the green?

Keith Mitchell was there with another shrink wrapped computer. It is amazing how much Keith can cram into almost any small space.

Jochen was there of course, but courtesy of Bernd Reinhardt, as his car is very poorly. Ditto Jochen, and we all wish him a satisfactory outcome to his own internal repair jobs.

Geoff Wicks (Just Words) was there enthusiastically, as always, expounding the virtues of his programs. I am especially grateful to him supplying me with a list of short English words. It was nice to see words like 'smsq' 'ql' 'mdv' and the like included by default.

Roy could almost have walked, if he had not had such a car full! He again had exactly the right serial lead for a customer of mine.

Bill Richardson had a good trading day, but returned with double quantity, having acquired two monitors. A renowned international QL show groupie was there fresh from a computer fire in Argentina, and bought Bill's entire stock of Z88s with the insurance money. I wonder whether the Gauloise were involved!

It was nice to see both John Taylor and Bill Newell still very much in evidence, despite not now being on the Quanta committee.

The AGM itself must have gone very well as it was over on around 30 minutes.

Quanta lives into its 20th year.

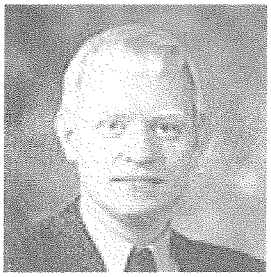
Food-fight? We heard there's some aggressive salad around!

Where are they now?

Simon N. Goodwin

Progress on UQLX is picking up again and I am planning to write the long-awaited feature for QL Today once Richard Z has finished some device-driver extensions (for QLSSS and hopefully MIDI) that I've been experimenting and corresponding with him about for more than a year.

I belatedly came across some information that might be of interest to QL Toady readers - apparently Ben Cheese has died. He was the engineer who got Sinclair's Microdrives to work. This is not fresh news - he died a couple of years ago - but as he made such a major contribution to the QL, I think it would be appropriate to note his passing, perhaps in a general 'where are they now' column. Here is some raw material with news of three QL luminaries. I attach the first pic of Ben I found on-line, but will try to scan better ones of him and "JM" over the next few days.



I only found out about his death while web-surfing for information about the ex-Sinclair team who I first met while writing about Flare, Konix and Atari consoles. I found references to his colleagues (who include John Mathieson, the "JM" of ROM fame mention in the latest QL Today, who now works for NVidia designing graphics cards) and went looking for the others - Martin Brennan and Ben Cheese.

An email from Andrew Owen, published online, described Ben as: "*an incredibly talented and even more incredibly nice chap who was one of the QL engineers. He also did mildly subversive cartoons for the Sinclair in-house newsletter (WHAM!, or What's Happening At Milton), and played saxophone with Shakatak, on one occasion.*"

The Norwegian Sinclair web site contains this quote, from one of their extracts from the Penguine paperback "Sinclair and the Sunrise technologies": "*it seems only fair to note that it was the tenacity and imagination of R&D staffer Ben Cheese that got the product to the market.*"

When Sinclair sold out to Amstrad, Ben formed a company called Flare with two other ex-Sinclair engineers, John Mathieson and Martin Brennan, who made the Flare 1 computer system, used in some arcade cabinets, documented in issue 11 (August 1988) of ACE magazine, later developed into the Konix Multisystem Slipstream prototype, then the unreleased Atari Panther, and finally returning to the mass market with the Atari Jaguar. Ben Cheese died on 21st January 2001, at the age of 46.

The Jaguar and Panther were both 68K-based, though the Flare custom chips delivered more horsepower in a far less easy-to-program form. The Panther featured three chips: a Motorola 68000 running at 16Mhz, an object processor called the Panther, and an Ensoniq sound processor called Otis, featuring 32 sound channels. The Panther could display 8,192 colours from a palette of 262,144, and could render 65,535 sprites of any size simultaneously. The Jaguar extended the custom 3D graphics hardware from 32 to 64 bits, but stuck with the 68000 as master processor - according to John Mathieson, "Atari were keen to use a 68K family device, and we looked closely at various members. We did actually build a couple of 68030 versions of the early beta developers systems, and for a while were going to use a 68020."

"However, this turned out too expensive. I always felt it was important to have some normal processor, to give developers a warm feeling when they start. The 68000 is inexpensive and does that job well." (from the Atari Jaguar online FAQ). So the QL experience had more implications than many realise! :-)

Further reading:

http://www.binarydinosaurs.co.uk/Museum/Sinclair/sinclair.htm
http://www.nvg.ntnu.no/sinclair/computers/peripherals/microdrive_sst.htm
http://www.homecomputer.de/pages/panther.html
http://www.digiserve.com/eescape/atari/Atari.shtml
http://www.wikipedia.org/wiki/Ben_Cheese

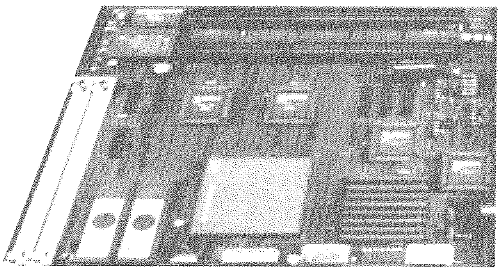
Incidentally Andrew Pennell, author of HiSoft MonQL, Devpac 68K and the Sunshine books Assembly Language Programming on the Sinclair QL, The Qdos Companion, and Master your ZX Microdrive, is now the lead programmer developing Microsoft Visual C++ in Seattle.

And your humble correspondent is now a Technology Programmer, writing programs to simulate water, heat and large animated graphics areas like grass and flowers on PlayStation 2 consoles, working for Attention To Detail in a barn in the Warwickshire countryside, with several other former QL programmers and rather more ex-Spectrum ones. Unlike the others, he's still actively writing on and about Qdos. :-)

Even ATD have an ex-Sinclair connection, as one of their earliest projects was to write system software for Flare before the launch of the Jaguar, and ATD went on to develop Cybermorph, the first Jaguar game. They still have a copy of Pennell's Qdos Companion in the company library!

Think your own thoughts.

Q60. The Super QL.



Features

- ✗ Q60/60 & Q60/66: 68060 CPU, 60/66 MHz, MMU+FPU
- ✗ Q60/80: 68LC060 CPU, 80 MHz, MMU (no FPU)
- ✗ 68060 superscalar architecture, dual execution units
- ✗ Up to 160 BogoMIPS performance for QDOS+SMSQ/E
- ✗ 16 to 128 MB RAM, PS/2 module sockets
- ✗ 256 kB ROM (mainboard supports up to 1024 kB)
- ✗ Highspeed 32 bit graphics + original QL hardware modes
- ✗ Up to 65536 colours at 1024 x 512 pixel resolution
- ✗ Multisync monitor output (15 pin HD connector)
- ✗ PC Keyboard interface (DIN)
- ✗ 20 kHz Stereo sound
- ✗ Battery buffered clock, 2 KB nonvolatile RAM
- ✗ Controller for 2 floppies and 2 IDE harddisks or CDROM
- ✗ 2 Serial ports with 115200 Baud, Parallel port (on I/O card supplied with mainboard)
- ✗ Hardware extension slot supports ISA cards
- ✗ Fits directly into AT Minitower or other standard case
- ✗ +5V / +12V power supply
- ✗ No tinkering, no parts from original QL needed
- ✗ Mainboard size 8.2 x 6.3 inch
- ✗ Can boot in a few seconds, directly from ROM
- ✗ Runs three different operating systems: SMSQ/E, QDOS Classic and Q60 Linux
- ✗ New „ShoeString" Q60 Linux distribution

Prices

Complete Systems	
Q60/60 Midi Tower*	
68060 @ 60 MHz, MMU+FPU, 64MB RAM, CD-ROM 56x, 3.5" Floppy, 20 GB Harddisk, Keyboard, 3 Button Mouse, 2 SER, 1 PAR, Stereo Sound	**£ 545.00**
Higher mainboard spec.	
Q60/66 (68060, 66 MHz)	**+ £ 139.00**
Extras	
RAM	
16 MB (giving 80 MB)	**£ 17.00**
64 MB (giving 128 MB)	**£ 36.00**
I/O Card (FLP,IDE,SER,PAR)	**£ 14.00**
Floppy disk drive	**£ 11.00**
Operating System	
OS programmed on ROMs**	**£ 10.00**
Q60 Linux CD	**£ 15.00**
Ethernet Card 10 Mbit/s	**£ 17.00**
Stereo speakers	
3 boxes, including sub-woofer	**£ 19.00**
Preinstalled software package	
QPAC1, QPAC2, FiFi, QD, PROWESS and much more, over £100 worth	**£ 59.00**

* Fully assembled and tested! Includes support disks and manuals.

** SMSQ/E and QDOS Classic available

Shipping and handling is extra. Prices may change due to semiconductor costs or exchange rates. Please note: Current SMSQ/E version supports only 16 MB out of 64 MB RAM, or 32 MB out of 80/128 MB RAM. Linux fully supports all memory configurations.

Q60

Q40

Website and technical information:

www.q40.de

Email: info@q40.de

D&D Systems

P.O. Box 5813, Ripley, Derbyshire, England DE5 9ZR
Tel. +44 (0)1773-740170, FAX +44 (0)1773-748399
After sales Tel. +44(0)1773-741164 (evenings)
Email: sales@q40.de

Financially assisted by a loan from QUANTA

Take the power back in your hands.

DISPLAY CODE Update - Part 2

Dilwyn Jones

Examples of Use

Following last month's listing and explanation, it may be more instructive to list a few simple examples of how to use these extensions for simple applications. Note that these routines use the Toolkit 2 extensions ALCHP and RECHP for allocating and deallocating temporary buffer areas in the common heap area of memory. Most systems these days have Toolkit 2 or equivalent commands so this should not be a problem.

1. Full Screen Window

This routine shows you how to set a window to occupy the full screen, or the full outline area available to it if it is a secondary window. To make channel #0 occupy the full area of the screen, enter the command FULL_SCREEN #0

If you tried the same thing on channel #2 on a VGA display, for example, #2 would be set to the maximum possible area as covered by the outline for the primary channel, in this case, #0, which would normally cover #0 and #1 and #2 in BASIC.

The procedure leaves the actual values in the variables dw and dh (width and height) and dx and dy (origin co-ordinates). Note that this routine doesn't actually do anything visibly, you may need to do a CLS command on the channel concerned, for example, to see its effect.

2. Store a Screen in Memory

This routine sets up an area in the common heap to store a copy of the current screen. The address of this area is given by the variable "area". We work out the start address of where to copy from the screen using the ADDRESS function, and the total number of bytes to copy is calculated by the product of the number of bytes per line (given by the BYTES function) and the height of the screen (given by the FLIM_H function). Note that when using 512x256 mode on the Aurora, for example, this routine will save the whole memory used for the screen, not just the visible area, as Aurora uses a fixed line length unrelated to the actual number of pixels used on the current display, meaning that although you only see 512 pixels across, for example, the line used to hold this diplay is 1024 bytes wide, but only 512 used and visible, so the calculation is not as obvious as might be thought at first. A typical application of this little routine might be to store a graphical screen in memory

```
100 REMark example routines for use with DISPLAY_CDE
110 REMark written by Dilwyn Jones, June 1997
120 :
1000 DEFine PROCedure FULL_SCREEN (channel)
1010 dw = FLIM_W(#channel) : dh = FLIM_H(#channel)
1020 dx = FLIM_X(#channel) : dy = FLIM_Y(#channel)
1030 WINDOW #channel,dw,dh,dx,dy
1040 END DEFine FULL_SCREEN
1050 :
1060 DEFine PROCedure STORE_A_SCREEN
1070 screen_length = BYTES(#0)*FLIM_H(#0)
1080 area = ALCHP(screen_length)
1090 IF area < 0 THEN PRINT #0,'Unable to reserve memory.' : RETurn
1100 MOVEMEM ADDRESS(#0) TO area, screen_length
1110 END DEFine STORE_A_SCREEN
1120 :
1130 DEFine PROCedure RESTORE_A_SCREEN
1140 IF area > 0 THEN
1150  MOVEMEM area TO ADDRESS(#0), screen_length
1160 END IF
1170 RECHP area
1180 END DEFine RESTORE_A_SCREEN
1190 :
1200 DEFine PROCedure MERGE_SCREEN (filename$)
1210 REMark merges a 32K 512x256 screen onto the top left corner of a
larger display
1220 area = ALCHP (32768)
1230 IF area < 0 THEN PRINT #0,'Unable to allocate memory.' : RETurn
1240 LBYTES filename$,area
1250 from_address = area
1260 to_address = ADDRESS(#0)
1270 FOR a = 1 TO 256
1280  MOVEMEM from_address TO to_address,128
1290  from_address = from_address + 128
1300  to_address = to_address + BYTES(#0)
1310 END FOR a
1320 RECHP area : REMark finished with it
1330 END DEFine MERGE_SCREEN
1340 :
1350 DEFine PROCedure FILL_MEM (addr,no_of_bytes,byte_value)
1360 POKE addr,byte_value
1370 MOVEMEM addr TO addr+1,no_of_bytes-1
1380 END DEFine FILL_MEM
1390 :
1400 DEFine FuNction SYSTEM_VALUE (what_offset)
1410 RETurn PEEK(SYS_VAR+what_offset)
1420 END DEFine SYSTEM_VALUE
1430 :
```

```
1440 DEFine PROCedure SET_MODE (mode_number)
1450  IF DMODE <> mode_number THEN MODE mode_number
1460 END DEFine SET_MODE
1470 :
1480 DEFine PROCedure IS_THERE_GD2
1490  IF GD2(#0) = 1 THEN
1500   PRINT'Colour drivers available!'
1510  ELSE
1520   PRINT'No colour drivers available.'
1530  END IF
1540 END DEFine IS_THERE_GD2
1550 :
1560 DEFine PROCedure VERSIONS
1570  PRINT 'S*BASIC    : ';VER$
1580  PRINT 'Operating system : ';OS_VER$
1590  PRINT 'Pointer interface : ';
1600  IF PTR_ENV(#0) THEN
1610   PRINT PTRVER$(#0)
1620   PRINT 'Window manager  : ';WMAVER$(#0)
1630  ELSE
1640   PRINT'None'
1650  END IF
1660 END DEFine VERSIONS
1670 :
1680 DEFine PROCedure IS_THERE_PE
1690  IF PTR_ENV(#0) = 1 THEN
1700   PRINT'Pointer Interface ';
1710   IF WIN_MAN(#0) = 1 THEN PRINT'and Window Manager '; : ELSE
PRINT'only ';
1720   PRINT'installed.'
1730  ELSE
1740   PRINT'No pointer environment installed.'
1750  END IF
1760 END DEFine IS_THERE_PE
```

while a menu is superimposed on the picture. The variable "screen_length" holds the actual length (in bytes) of the screen saved.

3. Restore a Screen from Memory

This routine restores the screen saved by the previous procedure, and releases the memory area used to store it, by using the RECHP command frm Toolkit 2.

4. Merge Screen

There is a large number of clip-art screens available for the QL, mostly as 512x256 QL screens. In the old days, when each QL had the same size screen, it was easy enough to load these direct to the screen with a simple

`LBYTES filename,131072`

command. Not only doesn't this work on modern large displays, it might actually crash the system in some cases, since the area of memory previously used by the screen may now be used by something else. This routine tackles this problem by loading the 32k (512x256 pixels) screens into a buffer area in the common heap, then copies it line by line into the top left corner of the display. Note how two variables are used to keep track of where each line starts. With old 512x256 screens, we know they are 128 bytes wide, so it is easy enough to step through them 128 bytes at a time. The other variable is incremented by the width of each display line, given by the BYTES function. Of course, writing direct to the screen is not the done thing, and the picture may well be ruined if another job is writing to the screen at the same time! Finally, when the transfer is complete, the heap memory is released with the RECHP command.

5. Fill Memory

A task which arises now and again in programming is to fill a given area of memory with a particular value. This routine takes advantage of how the MOVEMEM command works. The command should be issued in this form:

```
FILL_MEM start_address,
how_many_bytes, what_value
```

The routine works by setting the first byte of the area to be filled, using the POKE command. Then, it copies this up one byte with the MOVEMEM command, which repeatedly copies each byte up one address, thus the byte copied is always the value of the previous byte and the area is filled with the value of the first byte fairly quickly.

Another example: if you wished to turn the entire display black, then you could issue the following command. This is quite a naughty way of doing things, but it serves to illustrate how the command works:

```
FILL_MEM ADDRESS(#0),
BYTES(#0)*FLIM_H(#0),0
```

6. SYSTEM_VALUE

This routine reads a value from the system variables. You don't need to supply the absolute address, just the offset as published in several QL technical manuals. The routine adds the offset to the base address, peeks a value from there and returns it as the value of the function.

```
LET value =
SYSTEM_VALUE(offset)
```

For example,

`PRINT SYSTEM_VALUE (140)`

prints the value of the auto repeat delay, while

`PRINT SYSTEM_VALUE(55)`

prints the network station number.

7. Set MODE

Some programs which need to switch between 4 colour and 8 colour mode often set the screen mode (causing an irritating flashing) even if the screen was already in the required mode. A short routine like this can check the current mode and only change it if it is the wrong mode, thus preventing the flashing of windows you get when changing mode to the same mode.

8. Is there GD2?

Some programs with graphical content such as games or art programs may need to start up diffferently if the "colour drivers" (graphics device 2 or GD2) are present. The GD2 function makes this easy to check as this routine shows. So if a program found that it was running in QL 4 colour mode for example, but wanted to try to change to high colour, a simple test such as:

```
IF SCREEN_MODE = 4
THEN IF GD2(#0) = 1
THEN DISP_COLOUR 3 :
REMark switch to 16-bit
colour
END IF
```

9. Versions

Prints the version numbers of S(uper)BASIC, the operating system version number and the pointer interface version number. In this way, a program could if absolutely necessary check versions to see if recent enough for it to be able to run.

10. IS_THERE_PE

This procedure checks if pointer interface or window manager (or both) are installed. A simple application of this is for a pointer driven program to test if pointer environment is available and shut down tidily if not with a statement like

```
IF PTR_ENV(#0)=0 THEN
PRINT#0,"Sorry, I need
pointer environment to run"
PAUSE 50 : STOP END IF
```

[The full package will be available for downloading from my website on **http://homepages.tesco.net/dilwyn.jones/software/freeware/freeware.html** *for those who wish to avoid a lot of typing!]*

QL2K - Sinclair QL Emulation for Windows 2000/XP

Jimmy Montesinos

The port of Jan Venema's QLAY emulation

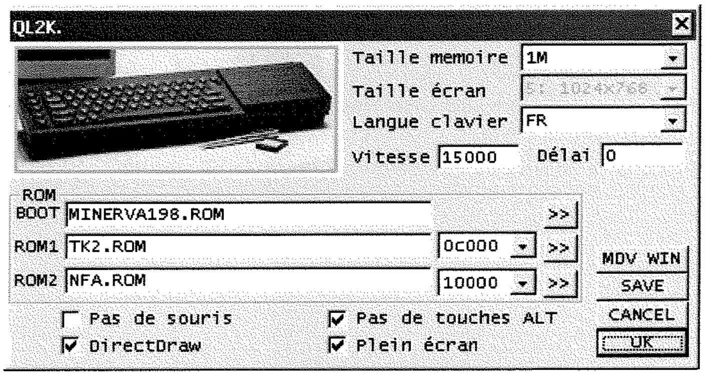
The design idea of QLAY was to fully simulate the Sinclair QL hardware and it is still downloadable from
http://web.inter.nl.net/hcc/A.Jaw.Venema
It was originally created to be run under Windows 95, DOS and Linux.

However, like many, Jan didn't have time to continue his development of QLAY, so we haven't seen a new version until now.

QL2K will try to continue the work done with QLAY but limited to new Windows machines such as Windows 2000 or Windows XP. It has also been successfully tested on Windows Millennium and Windows95.

In fact, QL2K is not only a simple port, as there are added features such as DirectX support and it will try to continue Jan Venema's development. In the same spirit, it will remain free of charge but registration is really needed to build a community of users. Why? - because if the development is done by only one person, QL2K cannot be translated to other languages and will not support all the QL ROMs that exist. This new community is communicating about problems they come across and support the development from day to day.

QL2K is certainly not the best QL emulation, probably a slow one at that, but some of you have told me there is a need for it. In fact, it is a very useful migration path for programs from the QL Classic Black box to QPC, Q40 or Q60 machines as some of these are too fast to provide a truly compatible QL system. Whilst transferring some of my old programs, I have really appreciated the delay option that permits me to localise pieces of faulty code in order to rewrite them properly. So, for some, the QL2K should be a very good development environment.


QL2K.

Taille memoire 1M
Taille écran 5: 1024x768
Langue clavier FR
Vitesse 15000 Délai 0

ROM
BOOT MINERVA198.ROM
ROM1 TK2.ROM 0c000
ROM2 NFA.ROM 10000

□ Pas de souris
☑ DirectDraw
☑ Pas de touches ALT
☑ Plein écran

QL2K is still in an early alpha version and debugging is in progress, but you can already use it. Corrections are being made as quickly as possible, and the next alpha, will support English, French and Spanish as well as Italian and German.

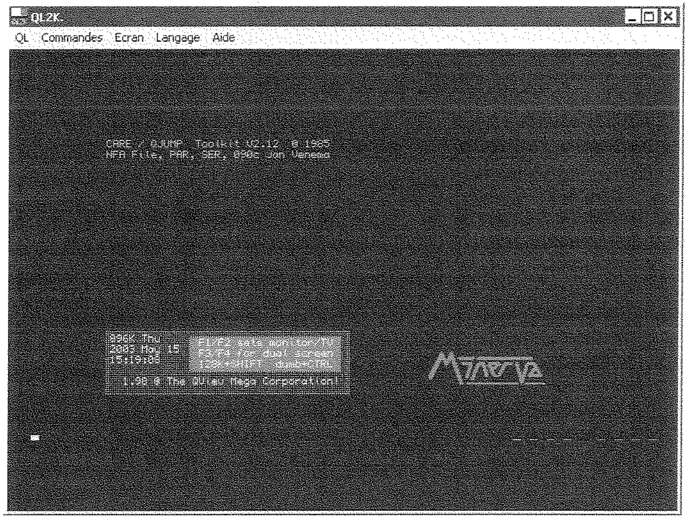
The UI has been redone, and you can configure several settings:

Memory size
Screen size
Keyboard Country
Speed Delay
Main ROM
2 extension ROMs
Mouse support
ALT Key support (the QL ALT key is re-mapped to the ALT-GR PC keyboard key as the ALT key is used for Windows and the QL2K menu system)
MDV_ drives (up to 8)
WIN_ (Qlay NFA) drives (up to 8)
DirectX use

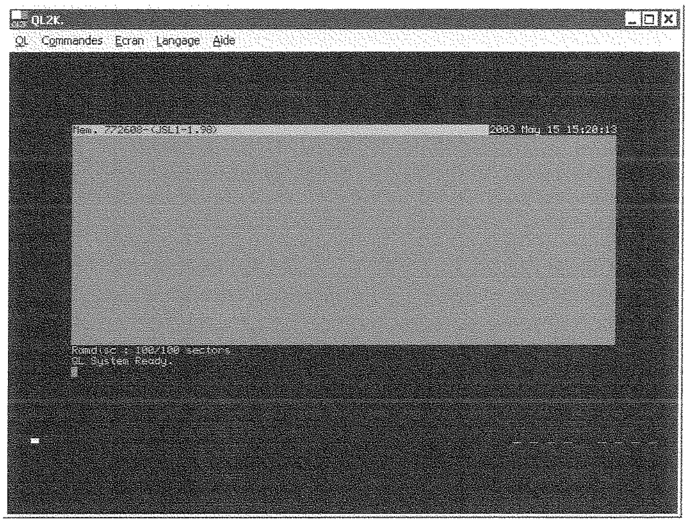
QL2K is hosted at
http://www.jadiam.org/QL/index.php
where you can download it. On the same site, you can access other pages including projects in progress and in the near future a FAQ and other QL2K support pages. I already have articles to publish in that section.

Moreover, you can write to me at **Jimmy.Montesinos@wanadoo.fr** for all concerning QL2K or simply the QL. And please, complete your registration form and email it to me in order to prevent QL2K becoming "abandonware".

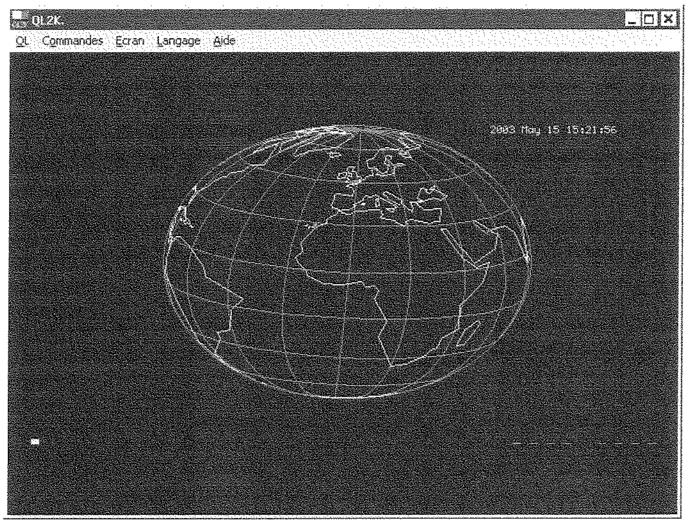
QL2K was at the QL US Show 2003 with the kindness of Phoebus Dokos, and already has many links onto several QL Websites. It will be also be added to the Emulator CD which can be found at RWAP Services and others.



```
CARE / QJUMP  Toolkit V2.12  © 1985
NFA File, PAR, SER, ©90c Jon Venema

896K Thu        F1/F2 sets monitor/TV
2003 May 15     F3/F4 for dual screen
15:19:08        128K+SHIFT  dumb+CTRL
  1.98 © The QView Mega Corporation!
```



```
Mem. 772608~(JSL1-1.98)                    2003 May 15 15:20:13

Ramdisc : 100/100 sectors
QL System Ready.
```



```
2003 May 15 15:21:56
```

GWASS and SMSQ

George Gwilt

There once was a fellow named Gwilt
Whose assembler was cunningly built
But they called him an ass
For trying to use GWASS
When Qmac fitted up to the hilt.

Now that the source code for SMSQ is available and I suggested that perhaps GWASS might be used to assemble it, I was given three reasons why that would not be sensible.

1. SMSQ source code is written for the assembler Qmac and the macros would all have to be re-written.

2. GWASS evaluates expressions in a non standard way incompatible with Qmac.

3. Anyway those intrepid souls aiming to alter SMSQ do not possess a 68020+, which is needed to run GWASS.

What follows is by way of an interim report, since, although I have not yet assembled every last file of SMSQ source code, I have assembled a large enough proportion of them to suggest that there would be no insuperable difficulty in using GWASS to assemble the lot.

First of all I should say that about six years ago I translated the macros supplied with QPTR for use with Qmac to assemble PE programs so that they could be used by GWASS. Indeed GWASS uses these adjusted macros to assemble itself. That experience was very useful when it came to amending the SMSQ source for GWASS.

It would be unusual for code written for one assembler to be capable of being assembled by another without some alteration. Although the mnemonics of the instructions themselves (such as MOVEQ #6,D0) may be identical there can still be differences. Some assemblers may require MOVEA when the resultant address is an address register while others may more leniently accept MOVE.

It is in the directives that more variation appears. Although DS, DC, DCB, EQU and EQUR are fairly universal, directives such as SET are less so. In fact Qmac uses SETNUM and SETSTR in place of the more normal SET.

It is of course in the macros that Qmac seems to distance itself most from other assemblers. Only at the simplest level is it orthodox with MACRO and ENDM heralding the start and finish of a macro definition. Differences appear as early as the definition of parameters. Qmac sets them by name after MACRO whereas the more normal procedure is to indicate their use inside the macro by \1, \2 and so on for the first and second etc parameters. Of course the strength of Qmac lies in the very many ways in which the parameters can be manipulated. Perhaps the most valuable of these is the ability to replace a parameter by its value. Qmac also allows a "goto" inside a macro which increases the power of Qmac's macros. These two facilities were added to GWASS six years ago.

Well, I knew in advance that all but the very simplest macros would have to be re-written, but what method should I use to tackle the general problem of altering the source code? With most people the only way to assemble source code meant for one machine on another is to alter the source code to comply with the second's requirements. In the case of GWASS, of course, I myself have another way. I can alter GWASS. In fact I have used both methods for the part of SMSQ that I have assembled so far.

One most important consideration when using a different assembler from the one originally intended is the possibility of wrong results occurring without any errors being signalled. This would especially be true if GWASS's original method of evaluating expressions were to be used.

For example

```
1 + 2*3
```

becomes 9 under GWASS's original method, whereas Qmac would produce the answer 7.

Just as a test I have asked various people what their answer would be. Most, of course, said "seven", but to my surprise both my wife and a granddaughter said "nine".

Anyway I have altered GWASS so that the option of using expression evaluation just like Qmac's is available. Other changes have also been made but they are relatively small and intended mainly to reduce the amount of change needed in the SMSQ source code.

I won't go into the details of these other changes made to GWASS. Instead I'll concentrate on the alterations needed in the source code given an updated GWASS.

(I hope that the new version of GWASS, v4.19, will be available by the time this article appears.)

Macro Calls

As I have indicated, the macros themselves require in most cases extensive revision. The macro calls on the other hand require less. But the need for changes does arise in the method of indicating parameters.

1. All parameters in a list must be separated by commas with no spaces in between. A space indicates the end of a parameter list, as does TAB or LF. However, any other character following a parameter will be taken as part of that parameter. Thus Qmac's assumption that a semi-colon immediately following a parameter is the start of a comment is not an assumption made by GWASS.

2. Parameters which may contain spaces are indicated by Qmac by curly brackets "{" and "}".

GWASS uses less than and greater than "‹" and "›" for this .

3. Comments on the line calling a macro are only allowed after the last parameter has been signalled. Thus a macro call with no parameters must not have a comment on that line otherwise the comment will be taken as a spurious parameter or the error "too many parameters" will be shown.
4. Missing parameters, signalled for Qmac by commas, must be signalled to GWASS by "‹›".

Relocatable Code

GWASS normally produces a type 1 binary file ready for CALLing, LRESPring or EXecuting. Only if the directive SECTION is given is SROFF relocatable code produced. Also SECTION must be issued befors XDEFs, XREFs, COMMENTs and so on.

In some SMSQ files SECTION is issued too late for GWASS.

Macros

The directory "mac" contains a set of files of macros. I have translated these and they can be used provided the necessary alterations are made to the calls to them.

One of the set is of a slightly different nature from the others. It is the file "creg" which deals with control registers. I have reduced the number of its macros from 20 to 7. This was possible because the underlying instructions can be assembled directly by GWASS.

Six macros which are still to be called in the same way as for Qmac are:

cena40
ciena40
cdena40
cdisa40
gcreg
pcreg

The remaining macro in the adjusted set is "screg" which now takes three parameters, the control register, the value to be inserted and the optional register to be used for the operation.

The macro "screg" is to be called instead of the nine:

scacr
iacr0
iacr1
dacr0

dacr1
itt0
itt1
dtt0
dtt1

Note that the first parameter of the new "screg" is the same as the old macro name for all except scacr. In that case the register name is cacr.

The remaining four macros are replaced directly as follows:

Old Macro (in place of the call)	New Instruction	
cpusha	cpusha	bc
cpushd	cpusha	dc
cinva	cinva	bc
cinvi	cinva	ic

Labels

In both Qmac and GWASS a label on its own takes the PC value at the first true instruction following it. That is, EQU, EQUR etc are ignored. The exception to this is that GWASS does not allow macro definitions between a label and its intended instruction. There is at least one SMSQ file where the initial label had to be moved later for correct assembly by GWASS.

Other Considerations

Although, as I have indicated, some changes have been made in GWASS so that fewer alterations are needed in the SMSQ source there remain incompatibilities. One of these is GWASS's 60-byte limit on input of strings. Another is the fact that the difference between two PC labels cannot be known until the whole program has been scanned even if the instructions at both places have been processed before the time that the value is wanted. This is because GWASS performs a compression of code which allows amongst other things the production of the shortest branches whether these are forwards or backwards. This contraction can alter the value of any PC label.

Although the definition and calling of macros form most of the places where alterations are needed, there are others. Qmac's SETNUM and SETSTR have to be replaced by GWASS's SET. Also the "value" convention indicated by square brackets is valid outside macros. Instances have to be located and altered.

Finally I notice that in many files macros are defined and used so that the macro problem is not confined to those in the directory "mac".

Final Comments

By now it must be as obvious to the reader as to the writer that GWASS could indeed assemble the SMSQ source. True, it would take me some more effort to produce the altered files needed – but I would hope by now not much.

Would it be worth it?

It would be worth it to me if I could put right the several niggling peculiarities in the SMSQ/E versions I use on my Q40 and Q60. One of these is the crashing of the machine if the BEEP parameter number 3(?) is given as 2(?).

And there are others. Oh yes! Why is there no shadow shown on PE applications under GD2? Not that I particularly want to see the shadow. It is more that I want to see if the shadow I have set in any window definition is really there!

Would it be worth it for someone else?

Given that production of the whole operating system consists of the two main stages of assembly and linking it seems to me that those who already have Qmac could assemble part of the system with Qmac and part using GWASS. Anyone wanting, for example, to use some of the 68020+ instructions for the operating system of a 68020+ machine could use GWASS for just those files. Although it is possible by using macros to set these advanced instructions coded by dc.w, it is very much more convenient to use an assembler which can do the job directly.

One example of usefulness is in the conversion of long integers to floating point. This is needed whenever the return parameter to a machine function is a long integer. A routine using the 68020+ instructions is superior to that using only the 68000/8 set.

Anyone interested in the alterations I have made to the SMSQ source for assembly by GWASS should contact me for details.

George Gwilt
39 Oxgangs Road
Edinburgh EH10 7BE

TEL/FAX +44 (0)131 445 1266
e_mail: geo.gwilt@argonet.co.uk

Programming in Sbasic with QPTR - Part 6

Wolfgang Lenerz

This time we continue to look at the level IV parameters used to make menu application subwindows - indeed the most daunting aspect of the Pointer Environment.

One of the first parameters is the rowlist, which we make with the ***MK_RWL*** (**M**a**K**e **R**o**W** **L**ist) function:

```
rowlist = MK_RWL (objlist,rows(n,1))
```

The parameters for MK_RWL are as follows (where "n" is the number of rows - 1):

-› * ***objlist*** is the "object list". This will have been obtained by the MK_AOL function, explained later in Level V.

-› * ***rows*** is an array DIM rows(n-1,1) where "n", as mentioned above, is the number of rows wished. This array is filled in by determining for each row, which object is to be the start object of the row and which is to be the end object. Let's suppose, for an example, that we wish our objects to be in 4 rows with three columns each, and in the order the objects are found. The array will thus contain: rows(0,1)=0 (start of row 1) and rows(0,1)=3 - end of row 0. The row 0 thus contains objects 0,1 and 2. Next, rows(1,0)=3 and rows(1,1)=6 - row 1 thus contains objects 3,4,5. As you can see, for the end marker we use the next element: to state that object 5 is the last object of row 1, we set rows(1,1) to 6.

  For each row, you MUST give as many objects as there are columns for the row! It is unfortunately not possible to specify simple x elements starting from y, where x is the number of columns. Nor is it possible, say in row 2, to have object 2, followed by object 18 followed by object 6. On the other hand, you can specify that row 1 has objects 5 to 7, row 2 has objects 1 to 3 and row 3 has objects 3 to 5 - overlapping is thus possible. The object number used here is simply their place (index) in the list of objects that you have built up in level V: the first object in the list is object number 0, the second is object 1 and so on.

The next parameters that need explaining are ***x_spacing*** and ***y_spacing***. They contain the "spacing list". This is obtained by the ***MK_ASL*** (**M**a**K**e **A**pplication subwindow **S**pacing **L**ist) function, as follows:

```
x_spacing = MK_ASL (size%(n,1), indsize%,
indspacg%)
```

where:

-› * ***Size%*** is an array DIM size%(m,1) where m is the number of columns (not rows!). For each element i, size%(i,0) contains the hitsize and size%(i,1) contains the spacing of object i-1.

-› * ***indsize%*** and ***indspacg%*** are optional parameters: they are used for the "index bars", something which nobody has ever really understood. They are best left at 0, even though they are explained later on.

Of course, defining one spacing list is not enough - we will only have defined the object sizes in one dimension (the x axis), but what about the other dimension, the y axis? Defining the spacing and size for one dimension is not sufficient, we know the sizes from left to right but not those up/down. So we must build a second spacing list, for the columns this time. This list is build up in a similar manner to the x spacing list above:

```
y_spacing = MK_ASL (size%(n,1),indsize2%,
indspacg2%)
```

where n is, this time, the number of rows minus 1.

Right, we have built the spacing list - now we have to establish the "control defintion". This tells the Window manager how many "sections" there are in the window (in each direction) and at what row (or column) each section starts.
A "section" is just a collection of rows (or columns) that can be scrolled independently. It is as if you cut the window into 2 (or 3,4,5...) independent parts, each with its own scroll arrows. Many windows only have one single section, but several are possible.
If all of your rows and columns fit inside the window at once you don't really need sections and, such a control definition isn't really useful and it can be left at 0. In Sbasic, the control definition also allows you to determine the colour and size of the scroll arrows, in addition to the sections themselves.
The control defintion is made with the ***MK_CDEF*** (**M**a**K**e **C**ontrol **DEF**inition) function:

```
x_ctrldef= MK_CDEF (secmax%,arrcol%,
barcol%,barseccol%)
```

x_ctrldef is then one of the parameters for **MK_APPW**.
The parameters for MK_CDEF are as follows:

-› * ***secmax%*** is the number of sections.

-› * ***arrcol%*** is the colour for the scroll arrows.

-› * ***barcol%*** and barseccol% are the colours of the index and section bars - again, leave these at 0.

With this, you have build a control defintion (here, for the x axis). The same applies if you want to have vertical sections:

```
y_ctrldef = MK_CDEF (secmax2%, arrcol2%,
barcol2%,barseccol2%)
```

Contrary to the spacing lists, you do not have to have a control definition for each dimension. If you do not have a control definition for any direction, the pointer may be left at 0.
Later, we will also have to initialise a definition control array, as follows:

```
DIM cty%(secmax%,2)
```

This will be used in the read pointer loop.
cty%(0,0) contains the number of current sections: Even if you have provided for the possibility of 2 sections, there may be only one to start with, or there may once have been two but the windows have been joined later.
cty%(0,1) is ‹› 0 if the control definition has changed.

Then, for each section i, elements (i,0), (i,1) and (i,2) remain. The contain the following information:
* (i,0) is the y position, in pixels, of the start of the section within the window.
* (i,1) contains the number of the start row (i.e. the first visible row).
* (i,2) contains the number of rows visible in the section.

Indexes

The two last parameters for MK_APPW, i.e. the x and y indexes concern the index bars, and they may be left at 0. If you do fill them in, they must contain the hitsize and spacing lists for the indexes (just like the ones for the window). Here are some details about the indexes. Menu application subwindows may be provided with "indexes" which are printed outside the menu application subwindow - for example the number of rows and columns in a spreadsheet. To do this, you must fill in all of the parameters concerning the indexes. I presume that Qspread (supplied by JMS) does use these indexes - and if it does, it must be the only application ever to have done so.

V - Level V: Defining the Object List

A - The object list

As we saw above, it behoves us to create an object list, which contains all of the objects of the menu application subwindow. This list is set up with the **MK_AOL** (**M**a**K**e **A**pplication subwindow **O**bject **L**ist) function.

```
objlist = MK_AOL(olattr, oljus%, olkey$,
oltype%,olstr$,olspr,olblb, olpat)
```

These parameters have the same meaning as for the **MK_LIL** function (see my earlier instalments in this series). However, there is no parameter defining the window or the sizes (we have already seen above how the sizes and spacings are defined). Nor do we define the origin of the object, which seems quite natural as the object is part of a regular and organised menu. Moreover, the object doesn't necessarily stay at a fixed position in the window, as it is possible to split an application subwindow into sections, and join them together later on. In addition, the menu may be scrolled or panned, and thus the object does not stay in a fixed position with respect to the window. However, we must define the attributes (same attributes for all objects) and then the justification, selection key, type and content for each object - these parameters should all be pretty clear by now. In the "files" subwindow of QPAC2, the type is of course a string and the content of the object is the name of the file. Actually, the type will generally be a string, but not necessarily so, as Jérôme Grimberts examples in these pages have shown!

B - "Blobs" and "patterns"

A **blob** is a structure that defines the shape of a visual object. This is similar to tracing a character on the screen: with a character editor, one can define what pixel must be "on" and what pixel must be "off". However, the character is only visible when it is printed on the screen with any INK on any PAPER (or, rather, STRIP). This is similar for blobs, except that you are not limited to the size of one character.

RWAP SOFTWARE

PWord English Dictionary v1.0	£15

The ultimate UK english dictionary with over HALF A MILLION WORDS, compiled by Paul Merdinian who compiled the Mega dictionary for DP's Speelchecker.
Due to the sheer size, a Super Gold Card is the minimum requirement.
Two versions are available: QTYP dictionary only on HD disk - £10 CD containing ASCII list, QTYP version, Solvit Plus (from Just Words) and the dictionary in the solvit format, cost £15.

QL Cash Trader v3.7	£5

A well established accounts package for the small to medium sized business, including automatic generation of profit & loss account, balance sheet, VAT returns, reports and analysis for audit trails and management decisions. Previously sold for over £100.*

QL Payroll v3.5	£5

Manage a payroll for a small to medium sized business. Handles up to 99 employees easily, producing P45s and P60s as well as the payslips on a monthly or weekly basis. Calculates tax and national insurance and is easy to update to take account of the current tax year rules.

Sidewriter v1.08	£10
Image D v1.03	£10
Q-Help v1.06	£10
Q-Index v1.05	£5

Four excellent programs to assist the QL user.
Sidewriter: Produce landscape printouts on Epson printers.
Image D: Produce 3D pictures of objects.
Q-Help: on-screen help for SuperBASIC commands.
Q-Index: look up keywords related to topics.
See earlier adverts for more detials.

ProForma ESC/P2 Drivers v1.04	£8

New improved colour and monochrome printer drivers, providing up to 720dpi for all programs written for use with ProWesS, such as LineDesign and Paragraph. Works on all Epson inkjet printers which support binary mode compression (740, 850 and 900 models at least). 1440 dpi to follow.

QL Genealogist v3.26	£20
Genealogy For Windows	£50

Store your family tree for posterity. Add individuals with details of their parents and children, watch all of those links build up into a formal family tree layout. Text files and pictures may also be linked to individuals as well as notes and events, making this the perfect way to preserve the history of your family.
QL version now supports FileInfo II and QMenu as well as allowing you to link both male and female trees.
Sample tree of the Royal family since 1066 included.
PC version is event driven - enter the details as they appear in documents and it generates the tree from these. QL data and GEDCOM can be transferred to the PC version. Upgrade to latest PC version (v5.21) for £8
Both programs easy to use and complete with a step by step tutorial.
** QL USERS upgrade to PC version for £25 ONLY **

Q-Word *COMING SOON*	£tba

The ultimate word game for the QL - you are given a grid of letters and need to link letters together to form as many words as possible. Points are based on both the number of words and their length, as well as letters used. As you use a letter, it is removed from the grid, with the object to clear the grid.
Using high colour graphics on all systems which support more than 8 colours (including Aurora), background music and much more, this will keep you entertained for a long time.

SBASIC/SuperBASIC Reference Manual	£40
Updates £6 each, £10 for 2 (Current Version - Rel 4)	

Have you ever tried to write a program, but been lost as to the means of performing a certain action? This Reference Manual provides you with a full description and examples of how to use all of the keywords found on each of the different QLs, plus SMSQ/e, Toolkit II and many different public domain toolkits. Details of any possible problems are provided, together with descriptions of how to use the device drivers and how to ensure that your programs are compatible across the range of QL platforms.
This book is ideal for all QL users and is kept up to date with regular updates. ** Currently Out of Print **

QL Cosmos v2.04	£5

Ever wondered what the stars in the sky looked like 100 years ago? Or, maybe you want to learn the constellations and names of what you see in the sky. This is the program for you - generates pictures of the stars and planets for any given place or time and provides details on these objects. Includes Halley's Comet, the Moon and the Solar System planets.

Q-Route v2.00	£25
Upgrade from v1.xx	£5

The latest version of this popular route finding program. Find the quickest route or the shortest route between any two places, using roads. A wide range of maps is available for this program (see elsewhere in this advert). The program is easy and quick to use. You can even add your own places and roads to the maps to include local detail.

Flashback SE v2.03 (Upgrade only)	£5

The ultimate database program - extremely fast and flexible, easy to use, updated to cope with the latest versions of the QL operating system and still maintained. A report module is included to allow you to format output in any way, including mail-merge. Unfortunately only available as an upgrade from the original version (original still available from Sector Software).

Return To Eden v3.08	£10
Nemesis MKII v2.03	£8
The Prawn v2.01	£8
Horrorday v3.1	£8
West v2.00	£5
The Lost Kingdom of Zkul v2.01	£5

A wealth of QL adventures - mainly text only.
Save the Galaxy from the ambitions of the evil dictator Nemesis.
Battle against werewolves and dracula look-alikes on a Hammer Horror set in the comical Horrorday.
Take the part of a prawn with a hangover, lost in a strange land in the hilarious Prawn.
Solve a bank-robbery by fighting the bad guys and collecting the loot in real-time old West.
Battle countless dwarves in the atmospheric Lost Kingdom of Zkul.
Return to Eden is a massive adventure over 3 disks with colourful graphics - control 3 characters in their quest to find the missing Prince.
All six adventures are available together for only £25.

A range of games to keep both the young and the young at heart amused. Some are old favourites, like Golf and a pub quiz program (500+ questions). Others are fast, colourful arcade games. Flight simulator also now available. Plenty of variation and skill required - what more can you ask for?
All 6 programs only £28.

Open Golf v5.20	£8
QuizMaster II v2.07	£5
Stone Raider II v2.00	£5
Hoverzone v1.2	£5
Deathstrike v1.5	£5
Flightdeck v1.05	£10

The latest maps for Q-Route. Maps of various areas of Britain have been created by cutting them out of Big Britain Map - they will use less memory and can contain more detail. Areas covered: Scotland, NE England, NW England, S&W Yorkshire, Wales & Derbyshire, London area and South England.
Latest version of Q-Route is recommended.

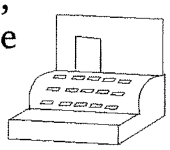
Britain.map v1.11	£2
BIG Britain Map (needs 2MB) v2.03	£5
Various Britain Area Maps (ask for details)	£2 ea.
Ireland Map v1.00	£5
Belgium Map v1.01	£2
Catalonia Map v1.03	£2

D-Day MKII v3.04	£10
Grey Wolf v1.8	£8
War In The East MKII v1.24	
(Upgrade Only)	£5

For the gaming enthusiast - D-Day is a classic table top wargame for one or two players - you control either the Allies or the Axis forces during WWII. With the ability to define your own army set ups and a choice of 4 different scenarios, this should keep you entertained for a while.
Grey Wolf is a graphical simulation of a submarine - can you sink the enemy shipping whilst avoiding their planes and destroyers??

* Also known as Trading Accounts

RWAP Software, 35 Chantry Croft,
Kinsley, Pontefract, West Yorkshire
WF9 5JH
TEL: 01977 610509
http://hometown.aol.co.uk/RWAPSoftware



Cheques in £sterling
payable to 'R.Mellor'

However, a blob has no colour, it just states that this pixels is on but not that another pixel is not on. It does not say what colour the pixel is to be, that will be defined by the colour pattern.

Thus, without a pattern, the pixel would be invisible, because it would be transparent, having no colour. The blob is like a mask which lets colour shine through or not.

A pattern is just the contrary – it is the definition of a structure with colours, but without a particular shape. By combining a blob (a shape without colour) and a pattern (a colour without a shape) we obtain something that is visible on the screen. A pattern without a blob can't be seen because it has no shape. Only the combination of the two produces something visible. A sprite is an example of a blob combined with a pattern, as it defines, at the same time, a shape and the colour of each pixel within that shape.

Let's re-use the example of the arrow which we had used for the sprite. It was something like this:

```
 90 DATA '   a   '
100 DATA '  awa  '
120 DATA ' awwwa '
130 DATA 'awawawa'
140 DATA '  awa  '
150 DATA '  awa  '
160 DATA '  awa  '
170 DATA '  awa  '
180 DATA '  aaa  '
```

This arrow can also be used as a blob because it defines a shape. It is just that for the blob, it makes no difference whether the colour in it is "a" or "w" or anything else. The only thing that counts is whether the pixel is transparent (' ' = off) or not (any colour definition – "w", "a", "r", "g" etc means that the pixel will be on). The colour itself is then filled in with the pattern. When the above data is used as a sprite, the pattern is made up from the colour information contained in the arrow data. But if the above is used as a blob, the colour information just tells us whether a pixel is on or off.

Now we shall apply a pattern to this blob:

```
DATA 'rrrarrr'
```

This mean that the arrow will be red, except for the pixel in the middle, which will be black. This pattern is applied to each row of the blob in turn and it is the combination of both that produces a visible object on the screen.

But why do it in such a complicated way when, as when have seen for sprites, everything could conveniently be made up in a single block? Well, that's just why: if everything is in a single block, you have to redefine everything if you want to change just one colour. If, for a sprite, I want everything to be red instead of black, I'd have the redefine the entire sprite. With a blob and a pattern, I just design the blob and several patterns and thus I can change colours as I want to, by using different patterns with the blob...

As a pattern may be defined in a single line, this is pretty fast! But a pattern may also be much more complicated and there may be one pattern line per line in a blob. This is what happens for sprites. In the above example with the arrow, QPTR makes up a blob and a pattern from the information contained in the data: a blob makes up the shape of the object, and there is a pattern with as many lines as there are lines in the blob.

For the basic programmer using QPTR, pattern and sprites are defined exactly like sprites – you should just make sure that the sprite origin is 0,0 because, of course, blobs and patterns don't have origin (and if you don't understand why not, even though a sprite has one, I'd recommend re-reading the section on sprites!)

You will be happy to know that this concludes the first big section of this series. By now, we have seen all there is about defining windows. In the next instalment, we'll be able to start on actually making the window appear on the screen.

Just a word of advice. I Have tried to cut up the window information into different levels, starting at the top level, and then working down. When you set up your window, you would, of course, do it the other way round: first you build the lower levels and then you work your way up, since you often need the lower level pointers and parameters for the higher level ones.

The User's Experience - the QDT Installer Example

Jim Hunkins

This article may be considered a progress report on one component of QDT. But I have decided to take advantage of my recent experiences in developing the QDT Installer to share some insights that I have on user interfaces.

It is probably safe to say that just about everyone in our QL community has experienced software that was too 'complex' to bother with, frustrating because things were hidden or not well documented, and just irritating because someone had thrown something together but never took a single minute to actually try to use it themselves, especially from an 'outsiders' viewpoint.

As I approached the idea of QDT the way that I prefer- a desktop to behave, I realized that this would easily allow me to fall into a trap where I assumed that everyone thought like I do (heaven help the world if that was true!).

Knowing better, I have constantly tried to stand back and try using pieces, looking for things that could be done more 'naturally'. Over the years I have paid some extra attention to this and have a big time appreciation for well done software and absolutely no tolerance for junk - especially when a great program is killed by a poor interface.

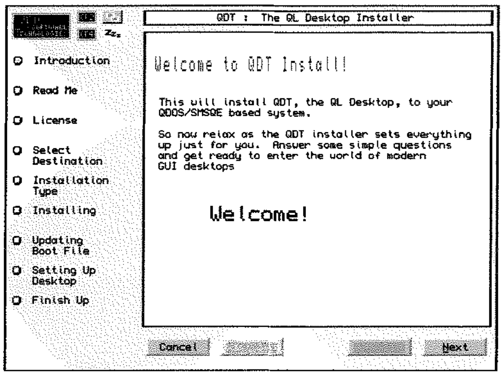
The Installer

With that all said, the first experience anyone has with a program is the installation. "As we all know, most of the time an installation amounts to copying some files, maybe updating the boot file, and having to read something.

For a desktop program with lots of files, lots of configuration, the last thing that I want to do is make someone have to go through a huge learning curve just to start using it. Therefore, taking examples from several nicely done installation programs, I have developed a full installer that copies everything over, asks a few basic questions, and then builds the user's first desktop, all automatically.

How to do this in a simple but informative way? It needs to be attractive, easy to understand, and definitely not confusing. This is the user's first experience and I want the user to survive it and be happily using their new desktop as quickly as possible.

Interface Tip: Presenting information can be cleanly done by breaking the screen into different, well thought out and consistently used sections. Looking at the intro screen, the QDT installer is broken into several such sections.



Welcome to QDT Install!

This will install QDT, the QL Desktop, to your QDOS/SMSQE based system.

So now relax as the QDT installer sets everything up just for you. Answer some simple questions and get ready to enter the world of modern GUI desktops

Welcome!

- *Standard Pointer Button/Program and Company Info:* this is arranged along the top
- *Progress/Section List:* the user can see every step that will occur or has happened. While it is a bit tough to see in the black and white image, in this screen capture to the left of the word 'Introduction', there is a colored arrow in the circle marking where the installer is at the moment.
- *Primary Activity Window:* this is the large light colored window where all the actual activity will be happening. This window is where the user sees messages, gives input, and sees progress reports. **Interface Tip:** By using a light, plane color versus the rest of the program screen, the user's focus is kept here.
- *Action Buttons:* along the bottom are four context sensitive buttons whose functions vary depending on what is happening in the program.

Interface Tip: You will note as you look at the images that the button functions are grouped and consistent from screen to screen. The left hand buttons are reserved for actions. The right hand group are the navigation for the program.

- the left one is always Cancel when available and is never used for any other function. This is one that you don't want someone to get confused and accidentally hit.
- the 2nd button from the left is used when specific extra options are available. For example, one screen sets it to allow the user to 'check' that their system is compliant with QDT's requirements.
- the right most button is the forward progress button. It normally says either Next or Start. In one case I use it for the license agreement. Basically it will always take the user to the next step.
- the last button (2nd from right) is also navigational but changes the direction/order that is taken. When available it will say either Prev (previous) or Skip.

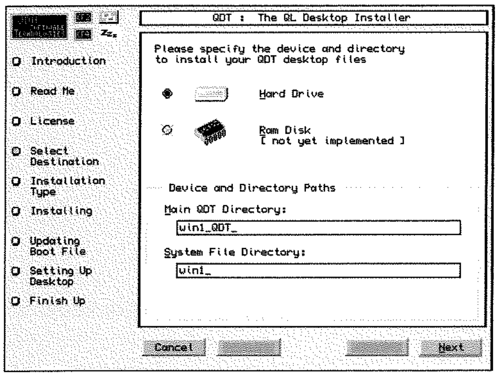
Sections

Introduction: the first thing that I want the user to read is a friendly hello. It shouldn't have too much information but basically set the expectations for what the program will be doing.

Read Me: for those users who will just not read a manual or pull up help screens until it is too late to actually help, this is for them. Here is where basic system requirements should go, any pertinent piece that would help (like, how to call up help while in the program), etc. It should be kept simple and basic. You want the user to read it don't you?

License: the importance of this should not be overlooked. You want to make it very clear what the user has purchased, what rights they have with the software, and any warranties or lack of warranties that you want to establish. While most people won't really read this at least to any detail, if you don't have it you are setting yourself for potential pain.
It is also important that you make it obvious that the user is agreeing to it. For this installer, there is an Agree button that has to be pressed twice, the second time in response to a question on the screen that leaves no doubt that they agree to the terms. With this installer, as with just about every other one worth two cents out there, if you don't agree, you don't install. Pretty simple but very important. The Gnu license agreement is a good general agreement example if you need one.

Select Destination: Ever have one of those programs that thinks that it knows the best place to put things? I have and absolutely hate it - suddenly you get tons of programs all stashed in one spot and can't find anything or know what went where.



QDT : The QL Desktop Installer

Please specify the device and directory
to install your QDT desktop files

● Hard Drive

○ Ram Disk
[not yet implemented]

Device and Directory Paths

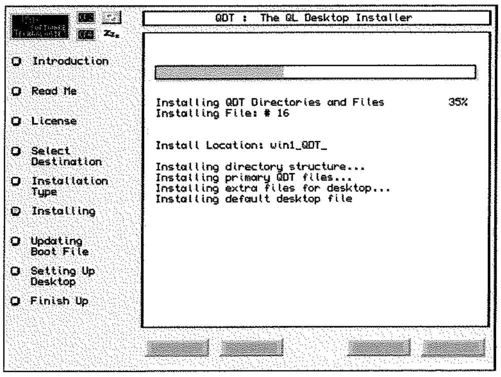
Main QDT Directory: win1_QDT_

System File Directory: win1_

Looking at the 'Select Destination' screen, you can see your choices clearly illustrated and obvious. The screen asks you to specify where you want to install the files. Since QDT will be optimized depending on what kind of device you use, you need to choose either the Hard Drive or the Ram Disk. And then you need to say which specific drive and directory to use, both for QDT and also for where most of your system files are. Since I can not guess at where users will put things, I have allowed flexibility.
When you first get to this screen, it won't look quite like what you see here. There will be no selection made, the Device and Directory will be blank, and only the Cancel button will be available. Once you choose the device type, the software fills in a 'guess' for the directories. But you can always change this guess.
In addition to the guess being filled in, once you have made the minimum selections on this screen, the Next button magically becomes enabled. Before the minimum information has been input the Next button is disabled and shows no text, preventing the user from getting too far without supplying the required input.
(Interface Tip) It is always nice to give the user options for how to make selections. For nearly every choice, the user can either click on the object or use a key stroke shortcut, indicated by an underline below the key letter. In this image you will note that there is an exception to this - the Cancel button. Since this is such a drastic and not normally used action, to avoid accidental selection, the Cancel button must be directly pressed. No shortcuts to mistakes here!

Installation Type: As with all good installers, the user is allowed different levels of interaction. The installer will happily do pretty much everything for the user, allow the user to check its actions, or allow the user to do it the hard way.

Installing: Now we get to the first very interesting piece. Looking at the Installing image, we see things finally happening. When I do something, I like to be informed. I prefer seeing things going on and really don't like sitting in the dark while something mysterious happens on my system. This screen is a good example of watching things happen.



QDT : The QL Desktop Installer

```
Installing QDT Directories and Files          35%
Installing File: # 16

Install Location: win1_QDT_

Installing directory structure...
Installing primary QDT files...
Installing extra files for desktop...
Installing default desktop file
```

Interface Tip: It is important that if you tell the user that progress is being made, that you don't lie or exaggerate. Many of you have sat there watching

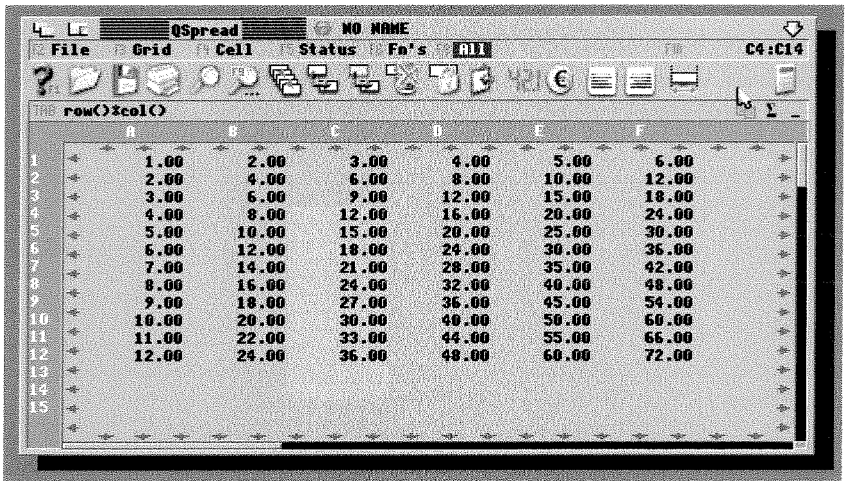
JOCHEN MERZ SOFTWARE

Im stillen Winkel 12 D-47169 Duisburg
Tel. 0203 502011 Fax 0203 502012
http://smsq.j-m-s.com

Updates, Updates...

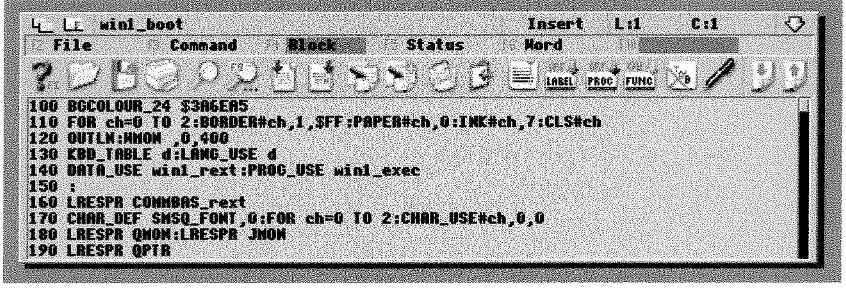
Well, we're getting closer now. It seems, I was a bit too optimistic, although the programs were more or less ready by the end of February. However, SMSQ/E is still changing (as you can see in Wolfgang's article in this issue), and we need to change the programs accordingly.

I have collected update sheets at the various shows, and will inform everybody as soon as the updates exist via email.



row()%col()

	A	B	C	D	E	F
1	1.00	2.00	3.00	4.00	5.00	6.00
2	2.00	4.00	6.00	8.00	10.00	12.00
3	3.00	6.00	9.00	12.00	15.00	18.00
4	4.00	8.00	12.00	16.00	20.00	24.00
5	5.00	10.00	15.00	20.00	25.00	30.00
6	6.00	12.00	18.00	24.00	30.00	36.00
7	7.00	14.00	21.00	28.00	35.00	42.00
8	8.00	16.00	24.00	32.00	40.00	48.00
9	9.00	18.00	27.00	36.00	45.00	54.00
10	10.00	20.00	30.00	40.00	50.00	60.00
11	11.00	22.00	33.00	44.00	55.00	66.00
12	12.00	24.00	36.00	48.00	60.00	72.00
13						
14						
15						



```
100 BGCOLOUR_24 $3A6EA5
110 FOR ch=0 TO 2:BORDER#ch,1,$FF:PAPER#ch,0:INK#ch,7:CLS#ch
120 OUTLN:WMON ,0,400
130 KBD_TABLE d:LANG_USE d
140 DATA_USE win1_rext:PROG_USE win1_exec
150 :
160 LRESPR COMMBAS_rext
170 CHAR_DEF SMSQ_FONT,0:FOR ch=0 TO 2:CHAR_USE#ch,0,0
180 LRESPR QMON:LRESPR JMON
190 LRESPR QPTR
```

The prices are set as well: Except for QD and QSpread, all other updates will be more or less free (provided, you own the "current" version). This means, owners of QPC 2 V3 will get the update for free, whereas owners of V1 or V2 will pay the normal upgrade fee as before.

There are still many things which need to be done (a new stand-alone WMAN and PTR_GEN is required, for example - for customers to run the products without SMSQ/E) but as most of you will have SMSQ/E we continue with the initial idea: you keep the old disk, and the new version will come on a new disk.

Therefore, a very minor charge for disk, label and postage will be all that you need to pay for Updates of QPAC2, QPAC1, WINED, FiFi, EPROM Manager, QMAKE etc. (we think it will work out to be about 1 EUR per disk incl. postage plus a basic 2 EUR for package). QD and QSpread upgrade will be 15 EUR each.

If you like, send in your masters now (as proof of purchase). You will get them back unmodified, plus the new version on a separate disk!

Of course, a news mail will be sent to the newsgroup as soon as things will be finally ready. But it is really close, I hope :-)

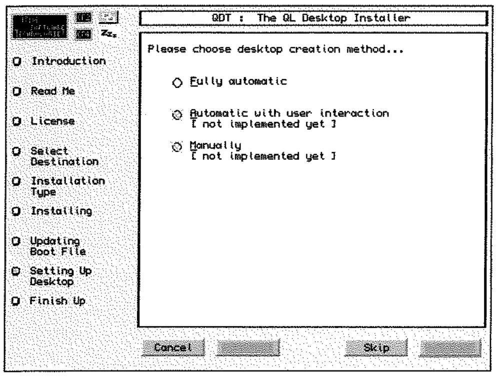
a progress bar zip ahead, thinking you were almost done. And then the bar just seems to stop when it was nearly at the end. Is the system dead, or is it just a poorly done progress bar and things are still going on? Should you abort the install or be patient? Not what I would call a good experience.

To be honest, progress bars are very hard to make accurate as things are rarely fully predictable. For this screen, I show different activities going on. The bar itself moves along fairly smoothly (looks very cool live and in color!). Since the actual installation pieces are very predictable (I pre-count the files to copy and know how many sections there are to be handled), I felt comfortable with giving a text percentage to go with it. A count of the files being installed is updated as appropriate, and each item that is happening is listed as it happens. This screen is constantly updating and the user won't have to wait for long for something to update.

Look at the buttons in this screen. Since action is still happening it would not be correct to allow the user to move ahead until the program is ready. Therefore, once the install starts, there is nothing that the user can do until it finishes. At that point the Cancel and Next buttons become active.

<u>**Updating Boot File:**</u> Whenever there is need to touch the boot file, there is always both a risk. And there are users who just aren't trusting enough (often for very good reasons). This screen gives the user the option of full boot file update (with a backup of the original file of course), or to have a copy made and updated, or to have just a list of updates necessary generated. I do believe in options but you will note that none of the choices that the user has to make at any point requires any special knowledge.

<u>**Setting Up Desktop:**</u> This is where I bet just about everyone will fall in love with this installer – through the use of the Fully Automatic method. If the user takes this option... Well, just wait and see.



QDT : The QL Desktop Installer

Please choose desktop creation method...

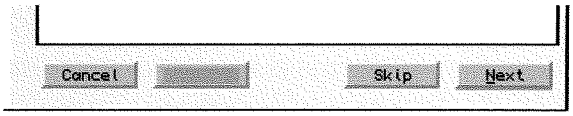
○ Fully automatic

○ Automatic with user interaction
[not implemented yet]

○ Manually
[not implemented yet]

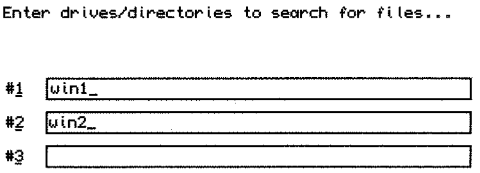
But before I show this off, this is a good place to illustrate how the button functions change. At this point you see both a Cancel button and one called Skip. Yes, if the user really wants to fully manually set up their desktop from scratch, picking Skip will setup an absolute minimum desktop (IE: the desktop icon). No default programs, no folders, nothing.

Once a selection is made, then the buttons will change. Looking at the next image, we see that the Next button has been added. Even more important, the Skip button is still there. No reason to assume that once a decision is made, it is the final one. This is why Cancel is there whenever possible.



This 'Setting up Desktop' section is unique. Before the desktop can be build there is still information to be gleaned. **Interface Tip:** For the sake of a clean and simple interface, this section is broken up into several screens, each totally independent of the others. Otherwise the screen would become cluttered, confusing, and potentially resulting in the user making a wrong selection.

The next partial screen capture shows the contents of the working window now changed to give a choice of drives that the installer can search for programs that could sit on the desktop. In trying to assist the user but not over do it, the System Directory from an earlier screen is set as the default entry into drive #1 slot. Of course it can be overridden. The other slots entries are left blank. There is really no way to guess what the user will want for these and many users won't want to search more than one drive. If the program had guessed and automatically put an entry into the 2nd and 3rd slots and the user didn't want anything, then the user would have to take extra steps to clear them. All around, for the average person this simple approach here is the best and will result in the minimum user's work.

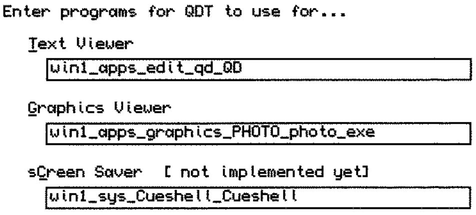


Enter drives/directories to search for files...

#1 win1_

#2 win2_

#3



Enter programs for QDT to use for...

Text Viewer
win1_apps_edit_qd_QD

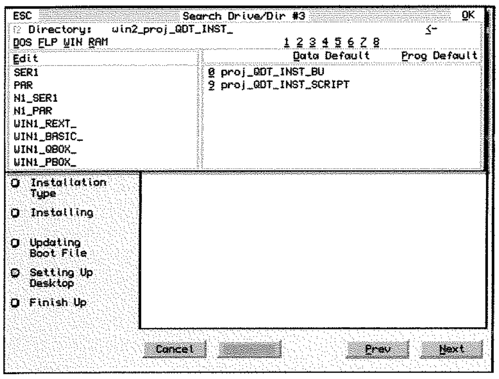
Graphics Viewer
win1_apps_graphics_PHOTO_photo_exe

sCreen Saver [not implemented yet]
win1_sys_Cueshell_Cueshell

Of course, any good desktop also needs

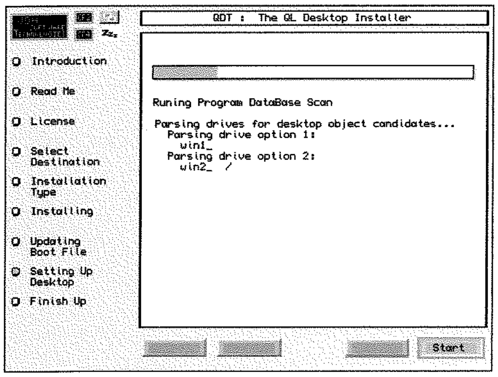
some default programs to handle certain types of files. No way to know what the user will want here either so these too start out blank. While the installer could definitely go into overkill here, the idea is to do the basic initial setup only. The user can later add more associations through the Property Notebooks or the FileInfo program (FileInfo will be directly used by QDT for this capability).

Interface Tip: It is nice to give a person different input options. A case in point here is how text entries are handled. If the user clicks on any text entry field with the left mouse button (space), then the text box is setup for direct keystroke input by typing/editing within the box. However, most people won't remember where certain files are. So it would be really nice to use the tools/interfaces available. In this case (image 8) QMenu is invoked for the user to be able to search and choose the actual file. In all text entries throughout the installer, the QMenu option is called up by either the right mouse button clicking on the field (Enter) or selecting the shortcut key for the field.



```
Directory:   win2_proj_QDT_INST_
SER1                 0 proj_QDT_INST_BU
PAR                  9 proj_QDT_INST_SCRIPT
N1_SER1
N1_PAR
WIN1_REXT_
WIN1_BASIC_
WIN1_QBOX_
WIN1_PBOX_
```

Now it gets even better and the installer does most of its work here. In the next screen capture, things start really happening. Remember how I said that it is very important to make the user feedback of progress mean something. Well, to be honest that isn't always completely possible In this case, the first part of building the desktop is to scan the drives (1 to 3, depending on the user preference). Obviously, it isn't possible to know just how long that will take.



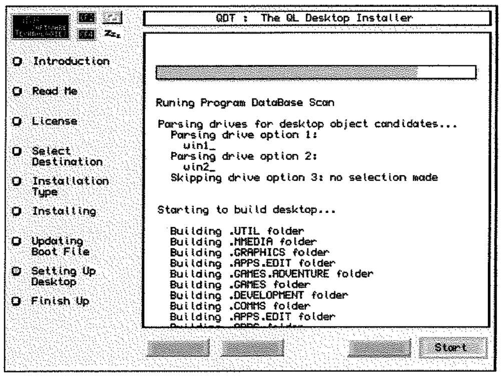
```
Runing Program DataBase Scan

Parsing drives for desktop object candidates...
  Parsing drive option 1:
    win1_
  Parsing drive option 2:
    win2_  /
```

For the desktop building I split the progress bar into a couple of pieces. The Running Program Database Scan was quick so it got the first 5% of the bar. Then the drives received anywhere from 30 to 45% of the bar depending if there was 1-2 drives to be scanned (30%) or 3 drives (45%). This combination was somewhat arbitrary but once the drives are scanned, it represents a significant amount of time has occurred. This bar usage break down felt right (I did experiment with different combinations - experimentation is often the key to how something looks and feels).

While a drive is being scanned, you can't guess how long that will take therefore the progress bar can't really move until the drive is completed. So how do you keep the user from thinking that things have crashed. I did it with a simple spinner, alternating between - \ | / characters. This can be seen the the right of the 'win2_' text. For every new directory that is scanned, the spinner changes to the next character. For the most part this gives the impression of something spinning on the screen and actually does indicate that something is still happening. There will still be 'short' exceptions. If a user keeps everything at the base level and doesn't use directories (then we need to have a long talk about disk organization :)) or in my case, a directory happens to be quite large, the spinner may seem to stall. However, I suspect that the stall period will rarely be for more than a second or two.

I did have a few options to make this more visually continuous. I had the option to advance the spinner for every file looked at but that does slow things down. I could have also spun a separate thread which sent an interrupt event back into the program to update the spinner - a lot of code for minimal improvement. The directory spinner advance was my best compromise for code speed and size versus user feedback.

The last screen for the 'Setting Up Desktop' section is where you can see the folders being built. This made for a couple more decisions. The first choice was should I split the screen to another one so that I wasn't displaying too much information on a single screen? The second choice was, if not, what about screen overflows.



```
Runing Program DataBase Scan

Parsing drives for desktop object candidates...
  Parsing drive option 1:
    win1_
  Parsing drive option 2:
    win2_
  Skipping drive option 3: no selection made


Starting to build desktop...

  Building .UTIL folder
  Building .MMEDIA folder
  Building .GRAPHICS folder
  Building .APPS.EDIT folder
  Building .GAMES.ADVENTURE folder
  Building .GAMES folder
  Building .DEVELOPMENT folder
  Building .COMMS folder
  Building .APPS.EDIT folder
```

I decided to keep everything on a single screen in this case as it all just fits together conceptually. It is good to see everything that is happening/has happened during such an installation. This way the user can do a quick review before going on. And since there is no input required there is no chance of the user getting confused and missing an entry. And if the user just doesn't care, he/she can simply ignore the screen progress updates.

My decision to handle the potential screen scrolling overflow had multiple parts. I decided to keep the progress bar totally on the screen of course (please never scroll an active element like this across a screen!). I also thought it was good to keep the drive info visible, including the fact that, in the particular install case, only one drive was chosen for scanning.

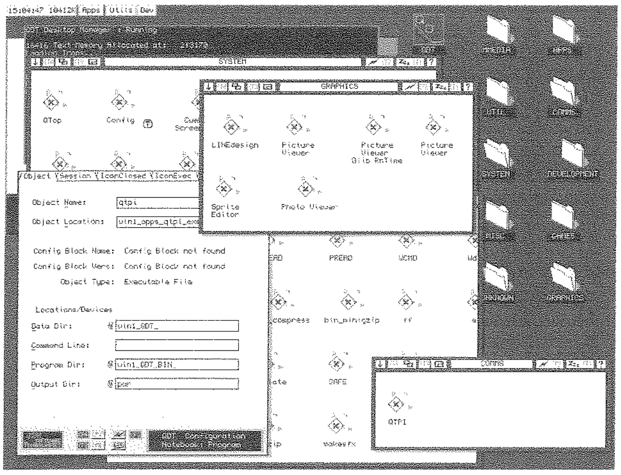
The folder building therefore was the only thing that got scrolled. For multiple reasons I ended up reverse scrolling into the window the new folders as they were built. Every new folder is shown directly below the 'Starting to build desktop...' line and other previously listed folders are scrolled towards the bottom of the screen. A bit different but I think the users will appreciate it once they see it in action.

For those of you who have really been following the user interface advise and studying the screen shots, you probably are jumping up and down right now. That depressed Start button shouldn't be there! And you are correct, and it won't be. This is a left over as I was tuning the user interface and moved all the progress screens to be their own separate screen. What you will eventually see will be four blank buttons, with the right one becoming 'Finish' after the desktop build has completed.

Finish Up: After all that is done, it is time to congratulate the user, give any final words of sage advise, and to... Or maybe not. This last screen will simply tell the user where everything is, how to start the desktop program itself, and to thank them for purchasing/trying QDT. Simple and neat.

And the Result Is!

Aha, so the program seemed to be doing something deep down in the computer. And indeed it has. In a test run, with it scanning my WIN1_ and WIN2_ drives, the desktop image that you see here was the result. The installer not only installed QDT but found every executable that I had and added them to folders. Using a database, it was able to fairly intelligently decide what went where. For those misfits/personal favorites that I haven't added to the database but show up as executables, they will be placed into the Unknown folder from which the user will be able to move or delete them as appropriate.



Object Session IconClosed IconExec

Object Name: qtpi

Object Location: win1_apps_qtpi_exe

Config Block Name: Config Block not found

Config Block Vers: Config Block not found

Object Type: Executable File

Locations/Devices

Data Dir: win1_QDT_

Command Line:

Program Dir: win1_QDT_BIN_

Output Dir: par

Now, all is not perfect (other than myself of course :)). Many, many programs will have the same name, or have different names on different computers. The installer's database helps by saying not to install programs such as 'config' since most are used one time only, along with tricks by avoiding the actual suffix and looking only at the primary name. It even knows that certain directories contain executables that you would not normally put on a desktop (like the printer files for some programs). Also some programs have suffixes indicating that they aren't a click and run.

For those programs that shouldn't normally be on the desktop or are placed in the incorrect folder, I can't do much but to assure you that the user will be able to delete, add, move around, rename, or whatever. The bulk of the dirty work is done at this point though and now it is the user's turn to tune their own desktop for how they want to work. The QDT user is ready to get started right away and take their time adjusting things as their time and/or desire allows.

Misc. Notes

Not bad for a short set of comments - I sometimes thing that I write like I talk, too much. With that said, there are a few additional notes to add here.

Let's imagine that during the install, something funny happened and you wanted to see just what went on. This will be especially useful if you need to contact me and tell me just how stupid the installer was (totally possible as I can't predict everyone's system setup). You could try to remember what happened from the screen. Or better yet, you could open up the installer log file and take a look (or send it directly to me. All major actions are recorded in a log file, including the case where if you have too many programs to fit into a folder. The log file will list the 'missed extras so that you can still decide to add them somewhere else. This actually happened on my unknown folder - it just tells me that I have to expand the database a lot still (thanks to those who wrote in before with their program database requests) and also clean up a bunch of left over junk on my hard drive (hmmm, perhaps I can modify and sell this as a Hard Drive Garbage collector).

I also realized while I was writing this article that I was breaking a very serious user interface rule. I forgot to make it obvious to the user as to how they should call up help for the installer. While to me it was very obvious that you just press the F1 key - hmmm, there is that old trap again - not everyone does think like me. So in the released version there will be a nice '?' in a button in the lower left corner.

For those of you wondering about the X's across some of the selection buttons, these indicate two things. One, this program is not done yet and the first testing version will be fully functional for the automatic installs but there are many options to be turned on over time. Two, since this is not the released version, some of the options/screens may move around a bit, especially as I hear some anticipated user feedback.

Final User Interface Tip: Before I sign off, my final and most important tip is to get user feed back, both during Beta testing and after releasing the programs that you are working on. It is amazing how good people can be in helping smooth out an interface.

Of course, after all this, since QDT is a labor of love (and definitely not profit), I hold the right to ignore any user's input if I so wish (IE: this is my desktop). Of course, please accept this last comment realizing the big grin that I have on my face as I type it.

More to come soon. Please check the QDT websight for updates and color screen shots. All these images in color and the new Icon Drawer color palette have been recently added.

Hints and Tips

Directory Devices

Dilwyn Jones

Sometimes it can be quite useful to know which devices are implemented on a particular system.

Here's a little BASIC program based on information from Ralf Biedermann in QL Technical Review many years ago. This lists device names on your system.

The routine shows how the directory devices list is set up. On my QPC2 system, for example, it lists:

```
DOS
DEV
FLP
WIN
RAM
```

Line 110 defaults the system variables to address 163840 (decimal) as they would be on an original QL. Minerva and SMSQ/E provide an extended VER$(-2) to return the system variables address, so lines 120 to 130 check if this is available. Line 140 looks in the system variables for the address of the directory devices list, then the "read_all" loop steps through the devices one by one, extracting the names and printing them to the screen.

Note: it really is not good practice to peek about in the system like this, as for example addresses might move in between lines of BASIC depending on what else is going on in the system. This routine has served me well, though, and should be OK for gentle tinkering on your own system!

```
100 REMark list directory devices
110 sysvar = 163840 : REMark locate
    system variables
120 v$ = VER$
130 IF v$ = 'JSL1' OR v$ = 'HBA' THEN
    sysvar = VER$(-2)
140 addr = PEEK_L(sysvar+72) : REMark
    directory drivers list
150 REPeat read_all
160   REMark pointer to length of device
      name
170   addr1 = addr+36
180   length = PEEK_W(addr1)
190   REMark get all characters of name
200   devicename$ = ''
210   FOR char = 2 TO length+1
220     devicename$ = devicename$
        &CHR$(PEEK(addr1+char))
230   END FOR char
240   PRINT devicename$
250   REMark next entry in linked list
260   addr = PEEK_L(addr)
270   REMark if pointer = 0 then end of
      list
280   IF addr = 0 THEN EXIT read_all
290 END REPeat read_all
```

Printing to an HP DeskJet with QPC2

Alf Kendall

For many years I have been using an Archive based program to track my stocks and shares. This is based on the set of procedures given in the black QL user guide but extended and modified over the years. While I was using the QL, Atari Mega and the Atari TT things worked just fine but problems crept in when I started using QPC. The print routines I had been using for years ceased to work as intended. Previously, when I pressed the "print" key the printer woke up, and the page prepared to print. For reasons which are beyond me the actual copy only came out of the DeskJet when the next page was sent to the printer or I deliberately sent a form feed command. With QPC2 v 3.03 it started the same but no page came out when the second page was prepared. Nothing happened with the next page but after the fourth page out popped a page containing the first three pages one after the other.

Marcel and Jochen could not help as they have no direct knowledge or access to a DeskJet. It seemed to me that there was some large buffer somewhere that was swallowing all the print output (and additional form feeds) and only passing it to the printer when it was full. To test this theory I modified my report writing program to it pad out with blank lines. Yes I had guessed right. If sufficient lines were added this "buffer" was filled up and the report procedure went back to normal.

In the report procedure I had already for other reasons included lines to count the output. The report had certain headers and footers which resulted in the total number of extra line to be printed to be 49. pq$ is the variable set by the print routine which directs the output to the screen or the printer.

The added lines were as follows:

```
pq$="y"
let padtot=(49-count)
while padtot>1
lprint "  "
let padtot=padtot-1
endwhile
endif
if pq$="y": return : endif
```

The printing system has now been restored to normal.

GD2 Colours Fills

Dilwyn Jones

In this article I hope to explain how to use the new colours available when using the so-called "colour drivers", or GD2 (Graphics Driver 2) to give the proper name.

GD2 has been designed to allow various colour schemes depending on the hardware on which SMSQ/E runs. To date, SMSQ/E with GD2 is already available for Q40, Q60, QXL and QPC2 that I know of, with a version for Aurora and Super Gold Card in development from Marcel Kilgus.

Please note: unless your hardware supports GD2 colour schemes, you will not be able to use the information in this article. Sadly, this includes QL users, since the original Sinclair QL and some emulators do not support extended colours.

The current versions of GD2 support 16-bit colour. In other words, dependent on the hardware, up to 65536 colours on Q40, Q60, QPC2 and QXL. The Marcel Kilgus development will offer 8-bit (256 colour) modes as well, for QPC2 and Aurora systems.

The GD2 specification documents from Tony Tebby refer to 4-bit (16 colour) modes and 24 bit (about 17 million colours, also known as "true colour") modes. As yet, there is no 24-bit colour SMSQ/E system that I know of, while the 16-colour mode was intended for systems like Aurora where, for example, the "flash-bit" used for flashing control in the 8 colour modes is used for extra colours, dependent on the system this might be 8 extra colours, or simply that the "flash" bit is used as an intensity control, giving the same eight colours in different shades. As far as I know, SMSQ/E does not currently have 16 colour mode drivers and no such support is likely to appear in the near future.

For this article, I'll concentrate on the 16-bit colour modes, as these are the most widely available modes at the moment.

Having decided on which colour modes we are using, there is then a choice of palettes and a choice of 4 colour definitions, just to complicate matters. Actually, although it appears complex, there is a very good reason for it.

The high colour modes can be set such that the old QL colour numbers stay the same, for compatibility with older programs for example. This is called COLOUR_QL, which is the name of the SBASIC command which defines which colour scheme is used. Here, colours 0-7 are the same as the old QL mode 8 colours. In essence, this is the old 4 colour mode with blue, magenta, cyan and yellow.

COLOUR_PAL is a palette mapped mode, where colour numbers from 0 to 255 can be defined by the user. So if you wanted some variety, you could redefine the colours - this may come in useful if you find you have programs which in theory use the standard QL mode 4 colours, but in practice some red appears as magenta, some white as yellow, some green as cyan. Thus you could redefine the "cyan" to be green, the "yellow" to appear as white, so instead of:

```
colour 0 = black
colour 1 = blue
colour 2 = red
colour 3 = magenta
colour 4 = green
colour 5 = cyan
colour 6 = yellow
colour 7 = white
```

we would redefine the colours (more on how to do this later) so that our list of colours is:

```
colour 0 = black
colour 1 = black
colour 2 = red
colour 3 = red
colour 4 = green
colour 5 = green
colour 6 = white
colour 7 = white
```

The COLOUR_24 command sets the "true colour" mode, where 24-bit colour values are accepted. There is no native support for this mode as yet, but the command is accepted in 16 bit colour modes, for example, and by the background colour command BGCOLOUR_24, where colours are approximated to the nearest colour in the current display type. For example, on QPC2 running in QL-style 4 colour mode,

```
BGCOLOUR_24 0+(256*255)+(256*256*0)
```

gives a green background.

COLOUR_NATIVE sets what is called the "native" colour mode. This is the actual colour system as used by the computer's hardware. In the case of QPC2 and QXL, the system is different to that used on the Q40 and Q60 due to the different organisation of the colour system hardware. On all 4 systems, 16 bit colour is used with individual bits controlling the red, green and blue components of the colour. On the Q40 and Q60, 15 of the 16 bits used to represent each pixel on the screen are used for the actual colour, and the sixteenth is used as an intensity bit, in effect doubling the number of colours from 32768 to 65536. On QPC2 and QXL, 5 bits are used for the red part of each pixel, 5 for the blue and 6 for the green. This basically gives twice the range of shades of green as for the other 2 colours. In terms of the values used by the MODE command, the QPC and QXL 16 bit mode is MODE 32, while the corresponding mode on Qx0 is MODE 33.

(For reference, the 24 bit true colour mode number would be 64, while the 256 colour 8 bit mode numbers would be 16 for the standard 256 colour mode with the colour value made up of 3 green bits, 2 red bits, 2 blue bits and a combined red/blue bit giving a composite value made up of grbgrbgx where x is the combined red/blue bit, and a mode value of 31 for the palette mapped 256 colour mode)

There is also a pair of commands called PALETTE_QL and PALETTE_8, which allows the colours to be redefined in QL colour schemes and 256 colour palette schemes respectively.

Programs generally default to the old QL colour scheme if started in high colour modes, which allows them to look as normal as possible in high colour mode. An old program which sets INK 7 for white ink, for example, may appear as a very dark blue, which is what colour value 7 looks like on a QPC2 system in 16 bit colour mode with native colour selected. Obviously, programs written to take advantage of the new colours would use native colour modes and use the appropriate new colour values.

The colour values are calculated as follows. If the colour is a QL or pallette mapped mode, a single value from 0 to 7 or 0 to 255 will suffice, as we have been used to with the old QL stippled colours for example. For 256 colour modes, the GD2 documentation files have a list which describes most of the colours, from colour 34 (pastel pink) to 54 (avocado) for example.

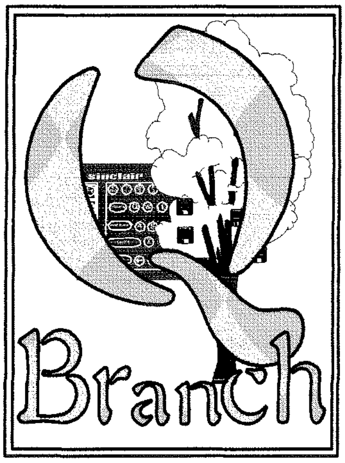
The colour value is calculated as a single number made up of the range of values allowed for the colour scheme in question, combined into a single 16 bit or 24 bit value. In some ways, 24 bit colour is easier than native colour as a whole byte (8 bits) is used for each component:

```
true_colour=(256*256*red)+(256*green)+(blue)
```

where of course "red", "green" and "blue" are the colour values, numbers from 0 to 255 each. The value of "true_colour" can then be used directly in an INK, PAPER, STRIP, or BLOCK command (from what I know, BORDER could only ever take 16 bit values). You may have noticed that colour is in fact stated as a 32-bit value, with 24 bits of this value used for the colour.

The most commonly used mode will probably be 16 bit native colour, and we will need to deal with the Q40/Q60 scheme and the QPC2/QXL schemes separately.

Q Branch



Text 87

£ 79.00

Typset 94	£ 29.00
Fountext 94	£ 39.00
2488 drivers	£ 29.00
Epson ESC/P2 drivers	£ 26.00

Text 87 is the only QDOS / SMSQ wordprocessor capable of handliing the full screen on the Aurora / QXL / QPC systems. New drivers are currently being written.

PROGRAMMING

QD 2003	£ 49.00
QD + QBasic	£ 63.00
QD + Qliberator + QBasic	£ 104.00
Qliberator	£ 50.00
Master Spy v 3.3	£ 30.00
QPTR	£ 32.00
Easyptr pt 1 & 2 (together)	£ 30.00
Easyptr pt 3 (C library)	£ 14.00
QMake	£ 18.00
QMon / JMon	£ 22.00
Basic Linker	£ 22.00
DISA 3	£ 34.00
QMenu	£ 16.00

Its Upgrade Time !

The new colour Window Manager is now part of SMSQ/E and it is time to upgrade your copies of the many programs which which have been converted to run with it. The really good news is that most of these upgrades are practically free.

Not all of these programs have been released to me yet but order the ones that you want and they will be sent on when they are available. All of the programs listed below are being converted to the new colour schemes.

These programs are free upgrades :

(Just send 75p for the disk and 80p postage + master as proof of ownership)

Agenda
QMake
Wined
FiFi II
QPAC 1
QPAC 2
QSup
All versions of SMSQ/E(Gold Card/QXL/Q40

These Programs are paid Upgrades:

QSpread 2001 to 2003	£ 10.50
QD 98 to 2003	£ 10.50
QPC2	-
update from v3.xx	Free
update from v2	£ 13.90
update from v1	£ 34.00

Return Master Disk With Order

QL Keyboard Membranes

Available Now !

£ 17.50 ea + P&P

UTILITIES

FiFi 2	£ 21.00
QSup	£ 30.00
QSpread 2003	£ 51.00
Cueshell 2	£ 15.00
Qload / Qref	£ 15.00
Disk Mate 5	£ 16.50
QPAC 1	£ 22.00
QPAC 2	£ 42.00
QTYP 2	£ 31.00
QLQ	£ 30.00

'Just Words' by Geoff Wicks

THESAURUS, STYLE CHECK

£ 10.00 ea / ANY 2 PROGRAMS £ 18.00 / ALL 3 PROGRAMS £ 25.00
(Includes Pointer and non-pointer driven versions)
(P.E. versions need Hot_rext, WMAN and PTR_GEN or SMSQ/E to run)
Upgrades from previous versions £ 2.50 + S.A.E. New Manuals £ 1.50

QL2PC

Convert text files from QL to PC formats and much more !

Only £ 10.00 Now with HTML support !

QL Rhymes £ 10.00

Rhyming Dictionary

Spelling Crib : PD program £ 1.50 +SAE
or Free if you buy all three programs

Tel +44 (0) 1273-386030 fax +44 (0) 1273-430501
Mobile +44 (0) 7836-745501
email qbranch@qbranch.demon.co.uk web : http://www.qbranch.demon.co.uk

Q Branch

Q Branch

Feeling out on a limb ?
Reach out for Q Branch.
Suppliers of Quality QDOS/SMSQ products
Hardware and Software.

20 LOCKS HILL, PORTSLADE,
E. SUSSEX. BN41 2LB. UK.

ProWesS

ProWesS (now free !)	£ 1.60
DATAdesign	£ 20.00
Fontutils	£ 28.00
File Search	£ 11.00
PFlist	£ 11.00
Dilwyn's Fontpack	£ Call
LINEdesign v 2.16	£ 22.00
PWfile	£ 17.50

Paragraph

The ProWesS word processor

¶ Demo version £ 1.50 + postage
Full Registered version £ 18.00

Version 2.03 available now !

Hardware

We have a small stock of second user items. Auroras / Qubides / Gold Cards / Qplanes / superHermes etc. call us to get details of the items available. These are going fast so call soon.

QXL II	£ 100.00*
Recycled superHermes	£ 65.00 *
Recycled Gold Card	£ 45.00 *
Recycled Aurora	£ 70.00**
Qubide	£ 50.00*
Qplane	£ 5.00
Aurora cables	£ 3.00
Aurora rom adaptor	£ 3.00
'Arfa Braquet'	£ 8.00
'Son of Braquet'	£ 18.00
The 'Braquet'	£ 16.00
MC plate	£ 6.50

* when available.

Qubide upgrades to version 2.01 £ 8.00

SMSQ/E

Gold Card / Atari / QXL Version

Now Only £ 32.00

Various Atari versions : call for details

QPC 2 v3.10 full colour version!

Special offer !
Get Cueshell for only £15.00 with any copy of SMSQ/E or QPC 2 upgrade !

Now Only £ 65.00

Q Branch Programs

The Knight Safe 3	£ 35.00
upgrades from previous versions	£ 5.00
Q - Count Pointer driven home accounting	£ 25.00
Q - Route v1.08C Route finding programme	£ 25.00
The Fractal Collection	£ 35.00

Post and Packing is included with Software Prices Unless Otherwise Stated.

VISA

We can accept payment by Visa, Mastercard and Switch. We also accept cheques drawn on a UK Bank in Sterling.
You can also pay directly into our bank in Euros.
Barclays Bank Acc# 79016888 sort code :20-71-02 Name :QBranch)

MasterCard

1. QPC2 and QXL

For these systems, 5 bits are used for red and blue, and 6 bits for green. This gives the following range of values for each of red, green and blue:

```
red - 0 to 31
green - 0 to 63
blue - 0 to 31
```

The combined colour value is calculated as follows:

```
native_colour = (2048*red)+(32*green)+(blue)
```

and this can then be used in an INK, PAPER, STRIP, BLOCK or BORDER command, e.g. INK native_colour

The listing in fig.1 (colours_bas) will allow you to experiment a little with the native colours on a QPC2 or QXL system. I'm afraid it only works on these systems, Q40 and Q60 owners will need to modify the range of values allowed for each colour. As I don't own one of these computers, I have not tried to rewrite it as I would be unable to test it.

Figure 1: COLOURS_BAS - a colour picker program.

```
100 REMark colour picker by Dilwyn Jones
110 :
120 COLOUR_NATIVE
130 PAPER #1,0 : CLS #1
140 red = 0 : green = 0 : blue = 0 : REMark RRRRRGGG GGGBBBBB
150 nat_col = 2048*red + 32*green + blue
160 REMark PAPER #0,nat_col : CLS #0
170 BLOCK 6*18-12,20,6,20,nat_col
180 AT 4,0 : PRINT nat_col TO 6;HEX$(nat_col,16) TO 12;BIN$(nat_col,16)
190 AT 0,0 : PAPER 0 : INK HEX('FFFFFF')
200 PRINT 'RED   GREEN BLUE'
210 element = 0 : REMark red
220 AT 19,0 : PRINT red TO 6;green TO 12;blue TO 18;
230 REPeat program
240   OVER #1,-1:BLOCK #1,30,10,36*element,190,HEX('FFFFFF'):OVER #1,0
250   BGET #0,key
260   OVER #1,-1:BLOCK #1,30,10,36*element,190,HEX('FFFFFF'):OVER #1,0
270   SELect ON key
280     =27  : EXIT program : REMark quit
290     =192 : REMark left
300            element = element - (element > 0)
310     =200 : REMark right
320            element = element + (element < 2)
330     =208 : REMark up, increase colour value
340            SELect ON element
350              =0 : REMark red
360                   red = red + (red<31)
370              =1 : REMark green
380                   green = green + (green<63)
390              =2 : REMark blue
400                   blue = blue + (blue<31)
410            END SELect
420     =216 : REMark down, decrease colour value
430            SELect ON element
440              =0 : REMark red
450                   red = red-(red>0)
460              =1 : REMark green
470                   green=green - (green>0)
480              =2 : REMark blue
490                   blue = blue - (blue>0)
500            END SELect
510   END SELect
520   BLOCK 12,31-red,36*0+12,190-(31),0
530   BLOCK 12,red,36*0+12,190-(red),BIN('1111100000000000')
540   BLOCK 12,63-green,36*1+12,190-(63),0
550   BLOCK 12,green,36*1+12,190-(green),BIN('11111100000')
560   BLOCK 12,31-blue,36*2+12,190-(31),0
570   BLOCK 12,blue,36*2+12,190-(blue),BIN('11111')
580   nat_col = 2048*red+32*green+blue
590   AT 4,0 : PRINT nat_col TO 6;HEX$(nat_col,16) TO 12;BIN$(nat_col,16)
600   BLOCK 6*18-12,20,6,20,nat_col
610   REMark PAPER #0,nat_col : CLS #0
620   AT 19,0 : PRINT red TO 6;green TO 12;blue TO 18;
630 END REPeat program
```

Figure 2 shows what this program looks like. It appears in window #1 and is controlled by the cursor arrow keys. You can set little sliders to change the red, green and blue components of a colour, which are shown as bars of red, green and blue, plus a block of the "native" colour. Beneath the block of native colour, the actual numeric value of this colour is shown in decimal, hex and binary, so that you can see how the value is calculated. I found this little program to be of immense value in helping me understand the colour schemes when I first started tinkering with the new colours.

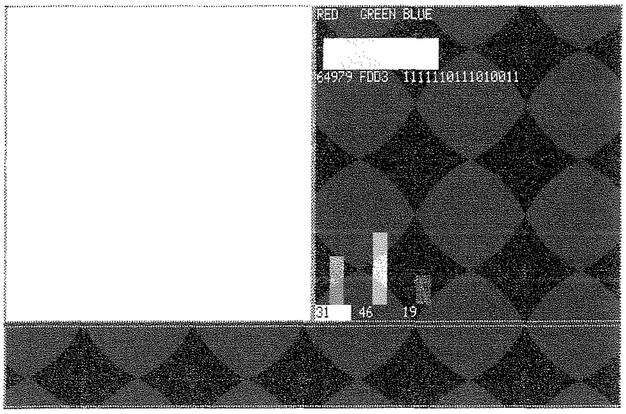


Figure 2 - The colour picker display

2. Q40 and Q60

The native 16 bit colour mode on these computers is a little different to QPC2 and QXL. For these computers, 5 bits each are used for the red, green and blue components, plus one bit used for intensity control.

```
red - values 0 to 31
green - values 0 to 31
blue - values 0 to 31
intensity - 0 or 1
```

So the combined native colour is calculated as:

```
native_colour = (2048*green)+(64*red)+
 (2*blue)+intensity
```

So your programming task for this issue is to modify the colour picker listing using the above information to work on a Q40 or Q60 system! (From here, Q40/Q60 are represented by the generalisation Qx0)

It might be useful for you to write a BASIC function to calculate the native colour in this way. The first is for QPC or QXL, the second is for Q40 or Q60:

```
DEFine FuNction QPC_NAT_COLOUR(red,green,blue)
RETurn (2048*red)+(32*green)+(blue)
END DEFine
```

```
DEFine FuNction Q40_NAT_COLOUR(red,green,blue,
 intensity)
RETurn (2048*green)+(64*red)+(2*blue)
 +(intensity)
END DEFine
```

Video Memory

A note may be appropriate on the video memory organisation for these systems. Experienced programmers may have worked this out already from the above information, but the colour scheme for the native colour values are represented at bit level by the following scheme, where r is red, g is green, b is blue and w is the 'whiteness' or intensity bit on a Qx0

QPC2/QXL

```
rrrrrggg gggbbbbb
```

The above is how the colour value is represented in an INK, PAPER, STRIP, BLOCK or BORDER command. The physical layout may not be the same. QPC2 for example looks byte-reversed compared to the above, the high byte and low byte seem to be the "wrong way round". It was probably designed this way because byte order of a word of memory is normally reversed on an Intel processor system compared to a 68000 processor system (I think the experts on this refer to this as endian-ness). The QXL is 100% compatible with this video memory layout. So the low byte of a word is at an even address and the high byte at the next higher (odd) address if you try PEEKing the values, and the colour bits look like this instead:

```
gggbbbbb rrrrrggg
```

(Unless you are naughtily peeking and poking into video memory you need not know this!)

Qx0

```
gggggrrr rrbbbbbw
```

is the colour word format and the video memory format follows this.

Some Examples of Colour

Black is black on all systems - i.e. no red, no green, no blue. A simple colour value of 0 will be black in all systems, unless colour 0 is redefined in a palette mapped system for example.

White is generally maximum of everything, again except in a palette mapped scheme. Consider QL mode 4 and 8 for example. In mode 4, each pixel had a green and red bit. Turn both on and you get white. In mode 8, turn red, green and blue bits on and you get white. Likewise, in true colour mode, a value of 255 for each colour component would give white, in this case 255 of red, 255 of green and 255 of blue.

In native colour modes, simply set each component to maximum and work out the 'composite' or 'combined' value. For a QXL, this would be 31 of red, 31 of blue and 63 of green, or a composite value of 2048*red+32*green+blue, or 65535 in decimal, or binary 1111 1111 1111 1111 or hexadecimal FFFF, whichever base makes it easier to visualise what's going on. On a Qx0 system, white would be 31 each of red, green and blue plus the whiteness or intensity bit, again all bits set, the same values as above for the QXL.

When you consider colours in between, things get a little bit more, umm, interesting (for "interesting" read "difficult" until you get used to it). Greys are easy enough, simply decrease the values of the component colours in proportion.
The next easiest is the primary colours of red, green and blue. If you remember some schoolboy physics, white light is made up of three primary colours and the other colours of the rainbow are made up of various amounts of these promary colours.
The strongest red is made up of just red, and maximum value (31 for all systems). Similarly, strongest blue would be just the same (31 on both systems). Strongest green would be green value of 63 on QPC/QXL or 31 on Qx0 (on Qx0 systems of course you'd need to consider the intensity bit). Therefore, native promary colours are as follows, with the decimal values in brackets and the preceding number showing how calculated. The Qx0 values assume the 'intensity' bit is set.

Colour	Qx0	QPC/QXL
red	64*31+1 (1985)	2048*31 (63488)
green	2048*31+1 (63489)	32*63 (2016)
blue	2*31+1 (63)	31 (31)

Your problem now is to work out what the corresponding values would be for the secondary colours of magenta, cyan and yellow. Hint: magenta is made up of blue and red, yellow consists of green and red and cyan is a combination of blue and green.

I have focused on native colours mainly because it's the hardest to use due to the unusual values for each promary colour, and partly because it helps to understand how the computer's hardware handles colours. You may find it easier to use COLOUR_24 and experiment to see how much easier it might be to be able to specify a byte sized value for each colour and get used to whether or not the computer behaves as you expect it to when you try to use COLOUR_24 but the hardware on the computer doesn't support that number of colours!

By now, if this is the first time you have tried to use the "colour drivers", you may be forgiven for having a headache trying to follow all this. The only real way to understand it is to actually use the new colours from SBASIC and experiment, experiment, experiment until you get your head around them. It took me quite a while to get used to them!
It is probably becoming clear that other colours and shades are made up of "in-between" values for each colour. Running the colour picker program listed above will help you to understand how various levels of red, green and blue alter the overall colour.

An interesting way to delve further into the new colours is to experiment with the BLOCK command (and other graphics) to 'graduate' colours, that is, draw small sections of colour side by side to give the impression of a smoothly changing colour. For example, since there are 32 possible values for blue (0 to 31) you could set up a block of colour with 32 different settings side by side, each with an increase in colour value of 1, to get a smoothly changing block of colour which graduates from black on one side to blue on the other.

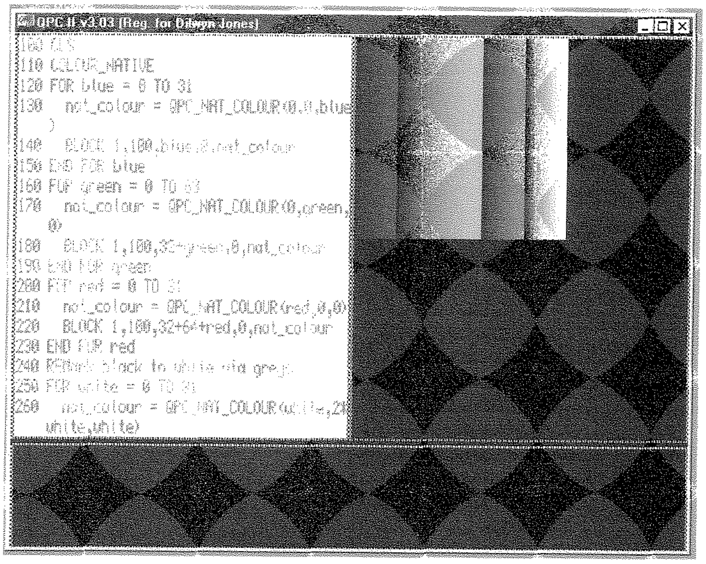
The listing in figure 3 shows one way of achieving this. It's designed for QPC and QXL, Qx0 users will need to use the other colour calculation function in place of QPC_NAT_COLOUR, and supply a value of 0 or 1 for the "intensity" parameter in each call to QPC_NAT_COLOUR. Since there are only 32 levels for green on Qx0 systems, change the loop counter in line 160 to count from 0 to 31 for Q40 or Q60, and change the 64 in the BLOCK statements to 32.

Each loop sets up a graduated block of colour from black to blue, black to green, black to red and black to white. You can of course get the secondary colours with suitable changes to the calls to the colour value calculation function - changing one of the zeros to the value for the second colour mixed in.

Figure 3 - a simple example of colour gradients

```
100 CLS
110 COLOUR_NATIVE
120 FOR blue = 0 TO 31
130   nat_colour = QPC_NAT_COLOUR(0,0,blue)
140   BLOCK 1,100,blue,0,nat_colour
150 END FOR blue
160 FOR green = 0 TO 63
170   nat_colour = QPC_NAT_COLOUR(0,green,0)
180   BLOCK 1,100,32+green,0,nat_colour
190 END FOR green
200 FOR red = 0 TO 31
210   nat_colour = QPC_NAT_COLOUR(red,0,0)
220   BLOCK 1,100,32+64+red,0,nat_colour
230 END FOR red
240 REMark black to white via greys
250 FOR white = 0 TO 31
260   nat_colour = QPC_NAT_COLOUR(white,2*white,white)
270   BLOCK 1,100,32+64+32+white,0,nat_colour
280 END FOR white
290 STOP
300 :
1000 DEFine FuNction QPC_NAT_COLOUR(red,green,blue)
1010   RETurn (2048*red)+(32*green)+(blue)
1020 END DEFine
1030 :
1040 DEFine FuNction Q40_NAT_COLOUR(red,green,blue,intensity)
1050   RETurn (2048*green)+(64*red)+(2*blue)+intensity
1060 END DEFine
```

Figure 4 is a screen dump of what to expect when the listing works, although as QL Today does not have colour...



```
100 CLS
110 COLOUR_NATIVE
120 FOR blue = 0 TO 31
130   nat_colour = QPC_NAT_COLOUR(0,0,blue
  )
140   BLOCK 1,100,blue,0,nat_colour
150 END FOR blue
160 FOR green = 0 TO 63
170   nat_colour = QPC_NAT_COLOUR(0,green,
  0)
180   BLOCK 1,100,32+green,0,nat_colour
190 END FOR green
200 FOR red = 0 TO 31
210   nat_colour = QPC_NAT_COLOUR(red,0,0)
220   BLOCK 1,100,32+64+red,0,nat_colour
230 END FOR red
240 REMark black to white via greys
250 FOR white = 0 TO 31
260   nat_colour = QPC_NAT_COLOUR(white,2*
  white,white)
```

Figure 4 - sample of output to expect

The output isn't especially spectacular or colourful. But it does show how a fairly simple loop in BASIC can create interesting colour schemes using fairly simple code thanks to the careful thought put in by the system designer, Tony Tebby.

One example of how such routines can improve your programs would be to use a similar routine to provide the equivalent of a CLS, but with say the top of the screen black and the bottom blue. Such routines are used a lot on other computers and can look quite nice as long as not overdone, so why shouldn't we have them!

The next routine provides a routine to clear window #1 as setup up for monitor mode, or after WMON 4 (i.e. clears a window 252 pixels wide and 200 pixels deep inside the border). Since the window is 200 pixels high, this means

Figure 5 - Graduated CLS for window #1

```
100 CLS
110 COLOUR_NATIVE
120 per_step = 32 / 200
130 FOR lne = 0 TO 199
140   nat_colour = QPC_NAT_COLOUR(0,0,lne*per_step)
150   BLOCK 252,1,0,lne,INT(nat_colour)
160 END FOR lne
```

```
170 STOP
180 :
1000 DEFine FuNction QPC_NAT_COLOUR(red,green,blue)
1010   RETurn (2048*red)+(32*green)+(blue)
1020 END DEFine
1030 :
1040 DEFine FuNction Q40_NAT_COLOUR(red,green,blue,intensity)
1050   RETurn (2048*green)+(64*red)+(2*blue)+intensity
1060 END DEFine
```

that there are more rows of pixels than there are shades of blue. We can take either of two approaches to work around this, we can work out how many pixels should be in each shade of colour (200/32), or we can work out how many units the colour would change for each pixel down the screen. This is what the program does in line 120, with the variable "per_step". If the window is larger than the number of available shades, this may well be a fraction less than 1 unit per pixel. If there are 200 pixels and 32 shades of a colour, this gives a calculated value of .16 unit of colour change per pixel row. This is OK, as long as we don't allow too many rounding errors to affect the colour calculation. I have put an INT statement to round down the colour value to the nearest whole number - omit the INT statement and run the program again to see what happens. You find that a few pixels at the bottom of the window appear black. What has happened is that rounding errors have made the colour value go slightly above 31, and what should be blue has in fact slightly spilled over into the next component of the colour value - the "black" bit at the bottom is in fact a very faint green on QPC/QXL. This demonstrates how care and an awareness of the limitations of floating point arithmetic is needed when handling calculations at this level.

To write general graduated colour block fills, you need to know how wide the block is, or how high the block is (depending on which way the colour change goes) and calculate the fractional steps involved, then step through the shape plotting BLOCKs in the appropriate colours. Once you have decided on the start and end colours, you can work out the differences between the two extremes and by how much the colour will change for each step. It sounds difficult, but basically only slight modifications to the above routines are needed:

```
per_step = block size / (end_colour-start_colour)
```

for each colour component. So as an exercise, write a procedure which allows you to fill a block of a user specified size starting with given colour and ending with given colour. The procedure definition will probably look something like this (this is only my untested guess-timate and no doubt you'll have great fun proving me wrong and developing and debugging this procedure!), you could also add x and y parameters for the top left corner of the block (in this case change the 0 and lne parmeters in the block to x and y+lne respectively.

Figure 6

```
DEF PROC BLOCK_FILL_DOWN(wide,high,red1,green1,blue1,red2,green2,blue2)
  green_step = (green2-green1)/high
  red_step = (red2-red1)/high
  blue_step = (blue2-blue1)/high
  for lne = 0 TO high-1
    nat_colour = QPC_NAT_COLOUR(INT(red1+(lne*red_step)),INT(green1+(lne*green_step)),
     INT(blue1+(lne*blue_step)) )
    BLOCK wide,1,0,lne,nat_colour
  end for lne
END DEFine
```

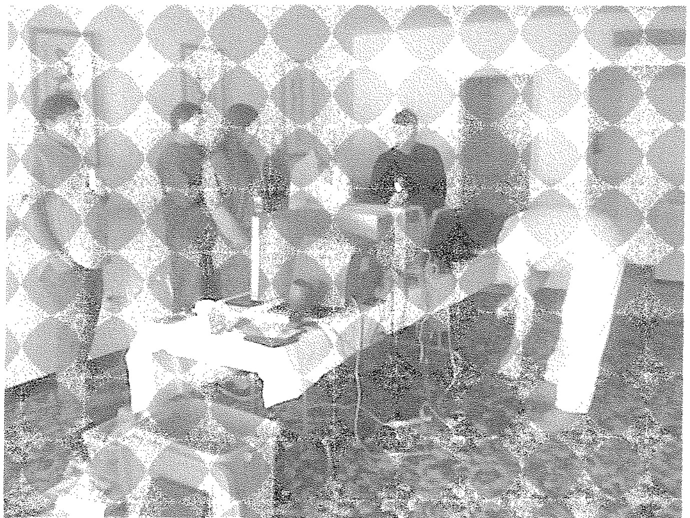
Of course, BLOCK isn't the only graphics command you can use. LINE, CIRCLE and so on all use the current INK colour. Experiment a little, I assure you that the reward is well worth the effort of mastering the new colours!

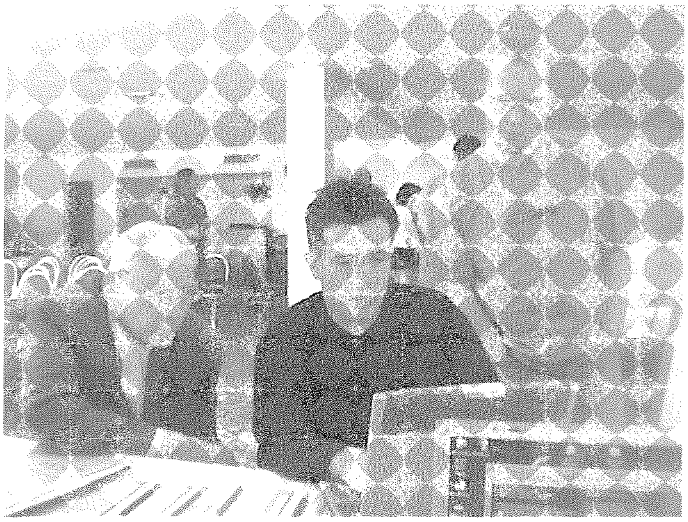
North American QL Show 2003

Roy Wood

This must have been my 7th visit to the US show. The first one was made in a flying hurry. I was convinced, by my previous experiences in trying to go the US for my work in the music business, that I would never get there so I booked a ticket just for the weekend. I flew out, got picked up - by Al Boehm not some loose woman I hasten to add - at the airport and was driven for several hours to get to the show. After the show I had a meal with the participants, went to bed in the hotel and then left the following afternoon. By the time I was standing at the luggage carousel waiting for my bags I wondered if I had left the UK at all. It was on that trip that I learned that Jochen often went off somewhere for a week and I have joined him in those jaunts ever since.

This trip began a bit shakily. We had always managed to plan our trips so that the flights arrived within an hour of each other. This year the general downturn in the air industry meant that there were fewer flights but it looked like we would all arrive together until United Airways cancelled the German flights and bumped Jochen and Marcel onto a later Lufthansa one. Last year's flight home for me was interrupted by an overbooked plane and I was awarded flight vouchers for the inconvenience. (Some inconvenience - I had flown from Jim's House on the West Coast and had to take a second flight from somewhere like Boston to get home. I wound up with a relaxing night in the Hilton and two free meals.) In fact I got more flight vouchers than the cost of this year's flight so I took my 21 year old son along with me. Our flight arrived early so we sat at the bar in the terminal where Jochen's was due to arrive and waited. We were there at 15.45 and, according to the last email I got from Jochen he was due at 21.30. US airports are not noted for their great restaurants, shopping facilities or general entertainment value so a six hour wait there was not an enticing prospect. Luckily Jochen had got it wrong (and also luckily I looked at the arrivals display or we would have still been in the bar when he walked out of immigration.) and found that his plane landed at 19.30. He might have found two people much the worse for drink had he arrived at the time he stated.



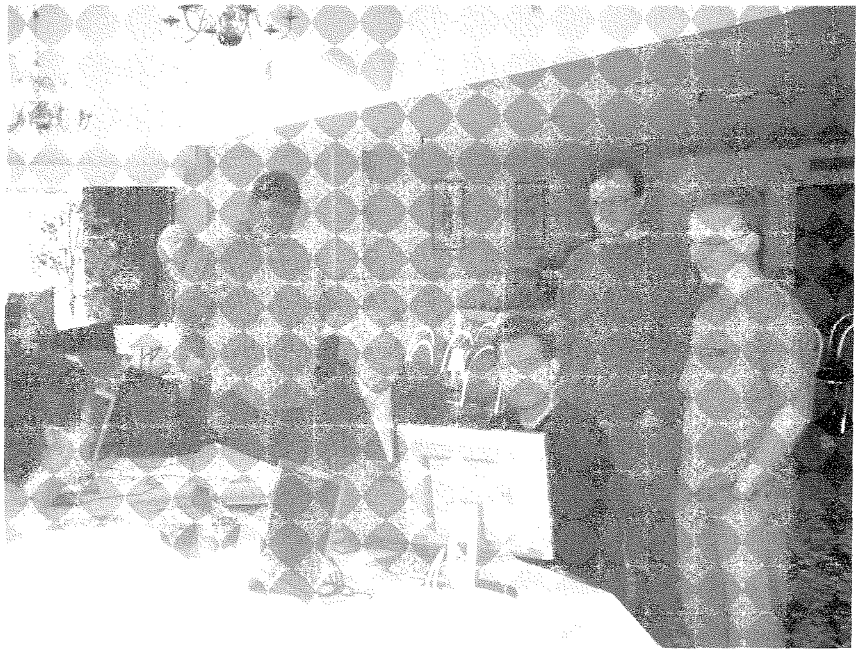


So, having collected the car and made the short (by US standards) trip to West Haven we were able to sink, exhausted, into our beds at the Econolodge.

The next morning we met for breakfast (or what passes for breakfast at these motels - a semi-stale, semi-bun thing with something which is almost exactly unlike butter accompanied by the incessant babble of airheads on breakfast TV.) and made our way up to the meeting room. We were missing two of our normal Euro contingent. Tony Firshman had a prior commitment and so could not make it (he was either singing in a choir or busking outside Tescos - not sure which) and Darren Brannagh has been very tied up of late having just achieved fatherhood. On the US side there were also some missing members. Ruth Fegley and Joe Lapunzina were unable to attend due to health reasons and were greatly missed. Jim Hunkins also had prior work commitments. Jim's absence was felt on two levels. We missed his cheerful smile and presence and he was also due to give a demonstration of QDT progress at the show.

One thing I had looked forward to was meeting Pheobus. He and I had exchanged emails for a long time and he has been very active in in the QL-scene in the last few years. He proved to be every bit as active in the flesh as in the cyber world darting from table to table, trying to solve Mel Laverne's problems with his tower cased Aurora and getting Herb to take his system apart. This was not too difficult because his system was just laid out in a straight line on the table - no case. I remember the first time I met him he had a monitor in a cardboard box. He bought it cheap because it had no case and he just stuck it in a box and cut holes for the cables. Once again the colour drivers were one of the show's highlights. We ran the latest versions of SMSQ/E on all platforms including Pheobus' dual Aurora / Q40 tower case. This did, in fact, show up some problems with the Aurora colour drivers when the system has a Qubide installed so Marcel had his laptop out trying to compile a version which ran.

One of the things that should have been seen at the show was, as I said, Jim Hunkins QDT. Jim had arranged to call in and do a remote demonstration over a speakerphone and to, that end, I had installed the latest version of QDT on the plane on the way over. Unfortunately the technology was not up to the task (i.e. Al Boehm left the microphone in his car and his wife had gone off with it) and so I had to be a surrogate speaker phone. This turned out to be a hard task - well multitask actually - because I had to hold the phone, operate the computer and attempt to turn Jim speak into something I could relay to the audience. It was probably none too thrilling to watch but we did manage to demonstrate the new install program and show how it picked



programs on the hard drive and made folders and icons for them. The 'not implemented yet' box was still quite full but the overall impression was that the program had advanced a lot since the first demos had appeared (for those who could not make it, there is an article in this issue!).

After the restaurant a small number of us headed out to a local bar for even more drinks in a very noisy atmosphere - great fun! Even better was the cabaret put on by some of the local, probably drunk, ladies - held my attention anyway.

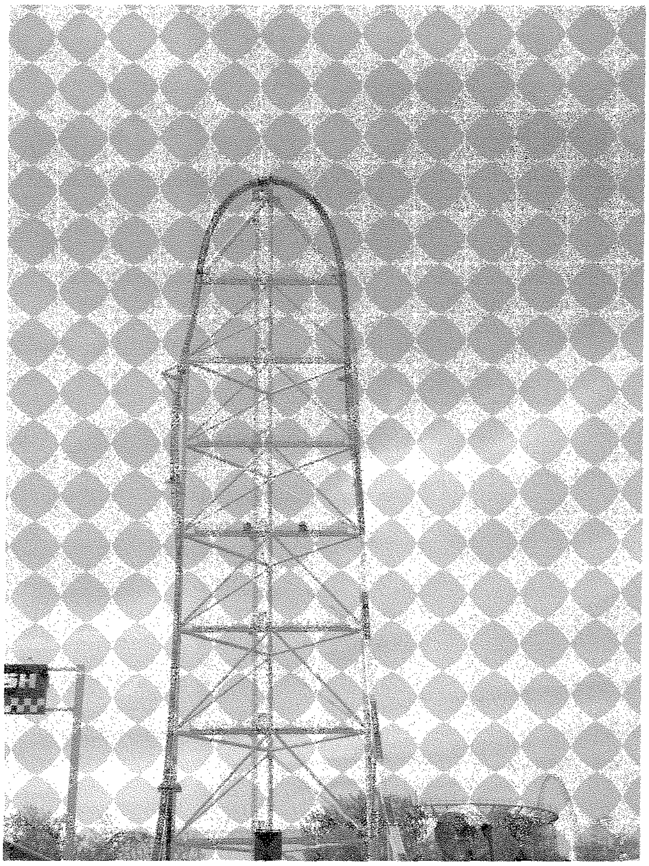
We parted company at breakfast the next day with the US users heading home and Jochen, Marcel, my son Tim and I facing an 11 hour drive to Cedar Point and the tallest, fastest Roller Coaster in the World (look at **www.cedarpoint.com** if you don't believe us). One advantage of these long drives is that I have a captive audience to bang on about changes I would like to see in some of the programs.

After my cabaret turn Pheobus demonstrated the new Q-Word game written by a partnership consisting of Rich Mellor for the code, Geoff Wicks for the word content and search engine and Pheobus for the stunning graphics. This is going to be a very popular program when it gets released.

During the lunch break Al Boehm chaired an impromptu meeting of NESQLUG (the New England Sinclair QL User Group although I always thought that sounded like an onomatopaeic description of someone sipping a hot chocolate drink.) The main item on the agenda was the proposal to change the name to NASQLUG (North America Sinclair User Group) and pull the various groups together under one umbrella.

After the show we adjourned to a local Mexican Restaurant (being a vegetarian I managed to divert them from the Steak House up the road) where a few margharitas were consumed with relish. They, once again, suggested that their next meeting be held in Europe but I have only one response to that - 'Don't do it guys' I need my excuse to travel to the US each year. Mind you, you are all more than welcome to attend one of our shows.

Next year the QL is 21 years old so let's see a whole lot more of you at the shows.



More shows to come this year: just look at the last two pages!
It would be great if we could see you all again at the shows - that's what they are there for! The QL shows should be encouraging for all visitors including the dealers, who often travel a long way to be personally available for you. A big thank you is also extended to the organisers for putting the shows on!

Scale Graphics

Dilwyn Jones

This article presents a very useful little function for use with SCALE graphics in BASIC on the QL. It should work with both SuperBASIC and SBASIC.

Recently I was writing a program which needed to use POINT, LINE and a few other graphics commands which use the BASIC graphics co-ordinate system. I needed to be able to scale the graphics concerned to fit the full screen irrespective of the screen resolution.

Modern QL systems offer a variety of screen shapes and proportions, for example, the QL in 512x256 and the Q40/Q60 in 1024x512 both have a 2:1 ratio of pixels across to pixels down the screen. But a QXL running in SVGA mode might have a resolution of 800 across by 600 pixels down, a ratio of 4:3 rather than 2:1. An Aurora or QPC computer running in EGA resolution of 650x350 would be a little less than 2:1 in ratio.

For reference, here are some of the more commonly used graphics resolutions on the various types of system I have used. You will notice that the list does not include emulators such as QLay, QemuLator and QDOS Classic as I do not have sufficient knowledge of those systems to be able to include them in this list.

Figure 1 - Common screen resolutions

Width	Height	Ratio	System type
256	256	1:1	QL mode 8
512	256	2:1	QL mode 4
512	384	4:3	QPC, Aurora
640	350	nearly 2:1	EGA mode on QXL, Aurora, QPC
640	480	4:3	VGA mode on QXL, Aurora, QPC
768	280	2.74:1	Extended 4 on ST-QL
800	600	4:3	SVGA on QXL, QPC etc
1024	512	2:1	Q40/Q60, QPC
1024	768	4:3	XGA on QPC

When a window is defined, the system gives it a SCALE value of 100 in the vertical direction, with the origin set at co-ordinate 0,0 in the bottom left hand corner. It is less easy to predict how many co-ordinates will exist in the horizontal direction - a window which is square in terms of number of pixels is not square in terms of graphics co-ordinates.

This gives us the problem that if we use LINE, CIRCLE, ELLIPSE etc to draw shapes and those shapes don't fit in the window concerned, part of the shape is drawn outside the window, although no error is caused thanks to the way the BASIC graphics commands work.

In fact, there is a relationship which is not very well documented which, if you know the window width and height and the vertical scale, allows you to work out how many co-ordinate points should be visible in the window concerned.

Figure 2 - The function listing

```
1000 DEFine FuNction X_Scale(y_scale,wide,high)
1010   RETurn 0.75 * y_scale * wide / high
1020 END DEFine X_Scale
```

The function takes three parameters. The first (y_scale) is the number of co-ordinates in the vertical plane. Unless you change it with a SCALE command it will normally be 100. If you have used a SCALE #1,150,0,0 for example it will be 150. The second parameter is the width of the window in pixels (if you are using QL TV mode windows #1 or #2 for example this might be 448, or 252 for QL monitor windows). The third parameter is the window height, normally 200 for both #1 and #2 windows in both TV and monitor modes on British QL systems.

Line 1010 uses this information to calculate the number of corresponding horizontal co-ordinates. Note how there is a relationship of 0.75 or 3:4 between horizontal and vertical. In other words, whatever the ratio of window width to height, the number of visible horizontal co-ordinates is 0.75 times the visible vertical co-ordinates times this ratio of pixels.

Basically, given the shape of the QL screen, introducing a scaling like this was necessary in order to make circles have a chance of appearing circular given that if you draw 'circles' with a 1:1 ratio of pixels they do not appear as circular on screen as QL screen pixels are not exactly square.

Here is a simple example of the use of this formula. How wide a line can we draw to fit in a window which covers the whole screen? We are working on a QL 512x256 screen and need to draw a circle halfway across the screen. The listing in Figure 3 (next page) shows how we might achieve this.

This might have seemed like a lot of fuss over nothing. In fact, coming up with this formula made it easy for me to write programs which scaled the graphics to fit the size and shape of the window concerned. Oh, which program was it? It was some of the screen saver modules in my LPsaver program (see news pages).

This routine is actually an approximation. You will find that partly due to the simplified calculations made and floating point rounding off errors etc, the actual scales may vary by 1 or 2 co-ordinates from what the routines indicate, but results are close enough for most purposes.

Figure 3: Using the function to centre a circle on screen

```
100 WINDOW 512,256,0,0 : CLS
110 x = X_Scale(100,512,256)
120 CIRCLE x/2,50,25
130 :
1000 DEFine FuNction X_Scale (y_scale,wide,high)
1010   RETurn .75 * y_scale*wide/high
1020 END DEFine X_Scale
```

New functionalities in SMSQ/E - Part 2

Wolfgang Lenerz

As promised last time, here is some information on the new colour formats and the system palette. SMSQ/E 3.00 is now released, you can get it from your usual supplier. Since there were some last minute changes, here is also some corrected info, so that you can have the latest documentation.

I - The colour formats

The new colour format, especially the 15 bit RGB format, has already been explained in these pages (several times,I think), so this is just a reminder. Colours are given as a word (16 bits), as follows:

`%00000000cccccccc`	- exactly as before
`%00000001pppppppp`	- palette
`%00000010pppppppp`	- system palette
`%00000011gggggggg`	- gray scale
`%00000100cc00tttd`	- 3d border (border calls only!) (see below - 2)
`%01ssxxxxxxyyyyyy`	- palette stipple (see below - 1)
`%1rrrrrgggggbbbbb`	- 15 bit RGB

1 - The stipple format

The stipple format for the palette stipple mentioned above is then as follows:

s = stipple code (0 = dot, 1 = horizontal,
 2 = vertical, 3 = checkers)
x = stipple colour
y = main colour

Of course, from this it can be seen that x and y can only hold 6 bits each. Thus, only the first 64 (2^6) entries of the palette can be used for stippling. However, due to the design of the palette, it is felt that those entries alone still cover the whole colour range quite well.

2 - 3d borders

The 3d border format is as follows:

`%00000100cc00tttd`

`d`= direction (0 = raised, 1 = lowered)
`t`= type
`c`= compatibility mode

The compatibility modes are available on some border types and they tell how to squeeze a non-standard border size into a QL border. Some modes paint areas with the current paper colour, therefore it is a wise idea always to set the paper colour before the border. The WMAN routines have already been changed to take this into account.

In case of a non-standard border width, any other border call on this window MUST be made through the WMAN routines instead of the standard border calls (e.g. by calling wm.trap3). Otherwise the overall window size will be altered. As a reminder, wm.trap3 is the one of the new WMAN vectors introduced by SMSQ/E 3.00 and it was explained last time (see QL Today, vol. 7 issue 6, page 35).

The colours to paint the border are defined in the system palette (sp.3ddark and sp.3dlight). Future versions may shade the paper colour, therefore it's again a good idea to set the paper colour before the border call.

II - The system palette

System palette handling has been extended and amended since the last time I reported on it.

There are now 4 (count'em) system palettes!

This means that a correction of the information given in the last QL Today is in order: in vectors WM.SETSP ($7c) and WM.GETSP ($80) D3 must contain the palette number (0 to 3) to which these vectors apply. Moreover, for $80, it is D2,a nd not D1 that is set to -1 to get the number of entries in the system palette. To avoid any difficulty, here are the two vectors, and their explanations, again:

```
Vector $7C                                                  WM.SETSP

     Set system palette entries

Call parameters                           Return parameters

D1.w start index                          D1   preserved
D2.w number of elements                   D2   preserved
D3.w palette number                       D3+  all preserved

A0                                        A0   preserved
A1   pointer to palette entries / 0       A1   preserved
A2                                        A2   preserved
A3                                        A3   preserved
A4                                        A4   preserved
A5   not used by any routine
A6   not used by any routine

Error returns:
     IPAR    Illegal index number / invalid number of elements
```

Set the entries of the system palette given in D3 (0-3)to the values in the buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1.

If A1 = 0 then the entries are taken out of the default table. Otherwise the buffer must hold an array of words with the colour values of the different items. The colour format is the standard WMAN colour format as described above.

```
Vector $80                                                  WM.GETSP

     Read system palette entries

Call parameters                           Return parameters

D1.w start index                          D1   preserved
D2.w number of elements / -1              D2.w preserved / item count
D3.w palette number                       D3+  all preserved

A0                                        A0   preserved
A1   pointer to entry buffer              A1   preserved
A2                                        A2   preserved
A3                                        A3   preserved
A4                                        A4   preserved
A5   not used by any routine
A6   not used by any routine

Error returns:
     IPAR    Illegal index number / invalid number of elements
```

Copies entries of the system palette into the given buffer, beginning with the index in D1 (counting from 0) and ending with the index D1 + D2 - 1. The buffer must be big enough to hold all requested entries.

If D2 is given as -1 the function just returns the number of items held in the system palette. This can increase when more items get defined in new WMAN version. This is guaranteed to be below 256.

When SMSQ/E is first started, the four system palettes are set up in such a way that they correspond to the older colourways 0 to 3. A job can thus easily use any of these to reflect the older way of doing things.

However, a user may change any of these system palettes by using the WMAN vectors (re-)described above. A job can also set up its own system palette. This is useful if you design a very nice colour scheme which doesn't correspond to any system palette and want to be able to preserve this for that job. Setting the system palette for a job is done this the WM.JBPAL vector (see last QL Today, page 36), wehere by you give the job the system palette to use in D3 If D3 is -1, then the A1 points to a palette that will be used only by this job.

III – Basic commands

The Basic programmer has not been forgotten and there are a number of keywords for palette and colour handling:

1 - Colours

The first of these are useful for colour handling. Their parameters are exactly the same as for the "normal" commands. The same is true with their names, except for the 'WM_' prefix:

`WM_PAPER [#channel],colour`

Sets the colour which is a word as described above. It also sets the strip as is the case with the normal PAPER command. But there is also the

`WM_STRIP [#channel],colour`

command to set the strip only.

Further commands are:

```
WM_INK [#channel],colour
WM_BORDER [#channel],width,colour
WM_BLOCK [#channel],xs,ys,xo,yo,colour
```

2 - Palette handling

There are commands to set/get the system palette and commands to set/get the per job palettes.

2.a - System palette keywords

`SP_RESET [#channel] [,number]`

This resets the colour palette given in number to the original values. Default is number 0.

`result% = SP_GETCOUNT`

Gets the number of elements contained in a system palette. Each system palette, of course, has the same number of elements.

`SP_GET [#channel,] [number,] address, first, count`

This gets the colours from a system palette and puts them somewhere. The optional "number" parameter tells us which system palette we want (0 to 3, default = 0). "address" is the address of the space for the information, "first" is the number of the first systemm palette colour to get (starting from 0) and "count" is the number of colours to get.

The space pointed to by "address" MUST have enough space for the number of colours! This is NOT checked by the keyword and it is the programmer's responsibility to make sure that this is so.

As an example, you could use the following code to get ALL of the colours of a system palette:

```
totcol%= SP_GETCOUNT
rem get nbr of colours in system palette
address= ALCHP(totcol%*2)+4
rem enough space for colours + security
first=0
SP_GET #1,0,address,first,totcol%
```

`SP_SET [#channel,] [number,] address, first, count`

Sets the system palette entries, the addres pointing to a space containing the colours. The parameters are similar to those for SP_GET.

2.b - Job palette keywords

`SP_JOBPAL [#channel], jobID/Job_name, number`

Set the system palette for the job given to the number. The job is given either as a string (e.g. "FiFi") or as a standard Job ID number.

`SP_JOBOWNPAL [#channel],jobID/Job_name, pal_pointer`

Set the job palette to the palette given in pal_pointer. Of course, the palette must have the format of a standard system palette.

And here is a reminder of the system palette content.

Name	Number	Meaning
sp.winbd	$0200	Window border
sp.winbg	$0201	Window background
sp.winfg	$0202	Window foreground
sp.winmg	$0203	Window middleground
sp.titlebg	$0204	Title background
sp.titletextbg	$0205	Title text background
sp.titlefg	$0206	Title foreground
sp.litemhigh	$0207	Loose item highlight
sp.litemavabg	$0208	Loose item available background
sp.litemavafg	$0209	Loose item available foreground
sp.litemselbg	$020a	Loose item selected background
sp.litemselfg	$020b	Loose item selected foreground
sp.litemunabg	$020c	Loose item unavailable background
sp.litemunafg	$020d	Loose item unavailable foreground
sp.infwinbd	$020e	Information window border
sp.infwinbg	$020f	Information window background
sp.infwinfg	$0210	Information window foreground
sp.infwinmg	$0211	Information window middleground
sp.subinfbd	$0212	Subsidiary information window border
sp.subinfbg	$0213	Subsidiary information window background
sp.subinffg	$0214	Subsidiary information window foreground
sp.subinfmg	$0215	Subsidiary information window middleground
sp.appbd	$0216	Application window border
sp.appbg	$0217	Application window background
sp.appfg	$0218	Application window foreground
sp.appmg	$0219	Application window middleground
sp.appihigh	$021a	Application window item highlight
sp.appiavabg	$021b	Application window item available background
sp.appiavafg	$021c	Application window item available foreground
sp.appiselbg	$021d	Application window item selected background
sp.appiselfg	$021e	Application window item selected foreground
sp.appiunabg	$021f	Application window item unavailable background
sp.appiunafg	$0220	Application window item unavailable foreground
sp.scrbar	$0221	Pan/scroll bar
sp.scrbarsec	$0222	Pan/scroll bar section
sp.scrbararr	$0223	Pan/scroll bar arrow
sp.buthigh	$0224	Button highlight
sp.butbd	$0225	Button border
sp.butbg	$0226	Button background
sp.butfg	$0227	Button foreground
sp.hintbd	$0228	Hint border
sp.hintbg	$0229	Hint background
sp.hintfg	$022a	Hint foreground
sp.hintmg	$022b	Hint middleground
sp.errbg	$022c	Error message background
sp.errfg	$022d	Error message foreground
sp.errmg	$022e	Error message middleground
sp.shaded	$022f	Shaded area
sp.3ddark	$0230	Dark 3D border shade
sp.3dlight	$0231	Light 3D border shade
sp.vertfill	$0232	Vertical area fill
sp.subtitbg	$0233	Subtitle background
sp.subtittxtbg	$0234	Subtitle text background
sp.subtitfg	$0235	Subtitle foreground
sp.mindexbg	$0236	Menu index background
sp.mindexfg	$0237	Menu index foreground
sp.separator	$0238	Seperator lines etc.

IV - Sprites

As mentioned last time, the sprite definition has been extensively modified, but in such a way that it should continue to be compatible with older sprites. However, there have been quite some changes since last time, thus I think it better to repeat the entire infomation here.

The sprite header is now as follows:

pto_form	$00	byte	sprite mode
	$01	byte	colour mode / system sprite number
pto_vers	$02	byte	dynamic sprite version number
pto_ctrl	$03	byte	sprite control
pto_xsiz	$04	word	X size
pto_ysiz	$06	word	Y size
pto_xorg	$08	word	X offset
pto_yorg	$0a	word	Y offset
pto_cpat	$0c	long	relative pointer to colour pattern
pto_mask	$10	long	relative pointer to mask/alpha channel
pto_nobj	$14	long	relative pointer to next object
pto_opts	$18	long	OPTIONAL (relative pointer to) options
pto_blk	$1c	long	OPTIONAL relative pointer to sprite block

A - Sprite MODE byte

The Sprite mode can be any of the following:

0	system sprite
1	traditional QL colour sprite (as before)
2	GD2 colour sprite

System sprite:
When the sprite mode is 0 for system sprites then the second byte is the number of the sprite. ALL other values are ignored in that case, i.e. a system sprite reference is only 2 bytes long.

QL colour sprite:
If this is a traditional Ql colour sprite, then the colour mode byte has one of the following values (as before)

0	mode 4
1	mode 8

GD2 colour sprite:

0	1 bit black&white
3	1 bit palette mapped
4	2 bit fixed gr palette
7	2 bit palette mapped
8	4 bit fixed irgb palette
15	4 bit palette mapped
16	8 bit fixed palette (equals Aurora palette)
31	8 bit palette mapped
32	16 bit QPC/QXL %gggbbbbbrrrrrggg format
33	16 bit Q40/Q60 %gggggrrrrrbbbbbw format
64	32 bit $RRGGBB00 format

B - Sprite control byte:

The sprite control bit is formatted as follows

```
%mpao0xcc
```

`cc` stands for a cache version number. Programs can increment this value to signal the cache that the sprite has changed. A special value is pto.fupd (force update, %11), which causes the system never to use the cached version. key: pto.cver

`a` flags whether the sprite uses an alpha channel instead of a mask (see below) key: pto.alph

`m` and `p` signal whether the pattern (p) or the mask (m) is compressed (see below) key: pto.pcmp and pto.mcmp

`x` signals that the optional pointer to a sprite block is present (bit x=1) or is not present (bit x=0) (see below) key pto..blk

`o` signals that options are present (=1) or not (=0) (see below) key pto..opt. This is currently unused but reserved for future use.

C - Alpha channel:

When the pto.alph flag is set in the sprite control byte, the mask is considered to be an "alpha channel". An alpha channel allows gradual mixes between the background and the sprite pattern. Every pixel is represented by exactly one byte. 0 means the pixel is completely transparent, 255 means the pixel is completely opaque. Values in between determine the degree of mixing of background and foreground. Alpha channel information is not padded at the end of each line. There's one byte for every pixel and nothing more.

D - RLE compression:

Both pattern and mask/alpha channel can be compressed using a simple RLE (run length encoding) algorithm. This is usefull with data that is largely homogene, which is often the case with masks.

Compressed data must be signaled in the sprite control byte (pto.pcmp, pto.mcmp) and starts with the bytes 'RLEx', with 'x' being either 1, 2 or 4. This is the item size the algorithm is working with. 8 bit RLE compression of 32 bit data wouldn't yield in good results, therefore the algoritm can also work on 16 bit or 32 bit data. After the ID there's one long word containing the size of the data in uncompressed form. After that the compressed data itself is following.

The compressed data always consists of one byte and one or more items. If the leading byte is in the range of 0<=x<128 then x+1 uncompressed items are following. Otherwise only one item is following, which represents 257-x times the same item in the uncompressed data. There are examples in the documentation that comes with the sources for SMSQ/E 3.00.

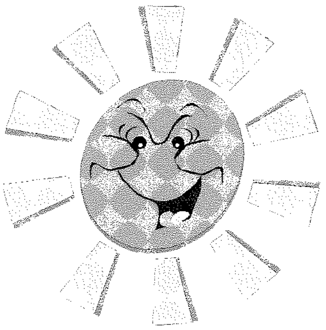
E - Sprite block

The object drawing routines have been amended so as to allow different sprites to be drawn in loose menu items, depending on the status of that item.

In such a case, it behoves the application to supply the different sprites and to set up a "sprite block" which is just a block of 5 longword pointers, as defined just below.

To keep things compatible with older versions of WMAN, this has been handled by setting a bit (pto..blk) in the sprite control byte. If this bit is set to 1, then there must be a pointer pointer to a sprite block. Please note that this pointer MUST be preceeded by the (pointer to) additional options (see section F below) which, currently is just a longword 0.

QL Today wishes you a great summer!



But please remember the deadline for articles! Summer-time is usually the time where we're really looking for your articles ... and by the time we finish this magazine there is only one article waiting to get into the next issue: the next part of Norman's assembler series.
It would be nice to have a few more program reviews - we know that there are not that many new programs around, but there are some. Q-WORD would be a good candidate! Then there will be the new versions of the 3D WMAN-style software ... hopefully out soon. And we have not had a review of the new WMAN on its own - programmers, users, how do you like them? Tell us and our readers!

Enjoy the summer!

IF pto..blk IS SET, THEN pto_opts MUST EXIST (AS A LONG WORD 0) AND pto_blk MUST POINT TO A VALID BLOCK.

The sprite pointer block is just a block of 5 longword pointers:

- pointer to sprite if item is available
- pointer to sprite if item is available AND is the current item
- pointer to sprite if item is selected
- pointer to sprite if item is selected AND is the current item
- pointer to sprite if item is unavailable

In all cases, these are long word relative pointers.

All but the first pointer may be 0. The first pointer (item available) MUST exist and point to a real sprite.

Null pointers are handled as follows
- For available items:
    * The pointer to the available item sprite MUST exist.
    * If no pointer to an available AND current item sprite exists, then the available item sprite is taken instead.
- For selected items:
    * If no pointer to a selected item exists, then the pointer to the selected item AND current item is ALSO ignored. The avilable item sprite is taken instead for both.
    * If no pointer to a selected AND current item sprite exists, then the selected item sprite is taken instead.
- For unavailable items, the available item sprite is used.

It is allowed, but not necessary, for any of these pointers, including the first pointer (available item) to point back to the original sprite, which will then be drawn as a normal sprite!

This allows three cases:

1 - The original sprite can be an ordinary QL mode sprite, which will be drawn normally by older versions of WMAN. The newer versions of WMAN will use the extended format.

2 - The original sprite could be a simple empty shell, with just the relevant data (bit pro..blk) and the pointer to a sprite block set.

3 - The original sprite could be a normal QL or 24 bit mode sprite which will be used by an item in any of its statusses.

Alternative 1 above will ensure that your software remains compatible with older versions of WMAN.

F - Options

To provide for future expandability of the sprite format, a special bit in the sprite control byte indicates whether the sprite definition is followed by an options long word or pointer. If bit pto..opt is set, then this options long word or pointer must exist.

At present, the options longword is unused. It has not been decided yet whether this will be a pointer to further options, or a bit map of further options.

Thus, for the time being, this longword can be ignored and even omitted UNLESS the sprite also contains a pointer to a sprite block (see above), in which case the options longword MUST EXIST (and, for the time being should be set to 0).

Byts of Wood

Saw Points Offcuts and Snippets

First off a small retraction

In the last issue I mentioned that the little utility that Microsoft's XP pops up when you have to close a recalcitrant program did not seem to actually do anything. Well, the other day I did manage to get it to do something. The machine was connected to the internet when it crashed and, since I was trying to do an update from M$ anyway I thought I would click on it and tell it to send the error message. It did redirect the browser to a help site. Trouble was it gave the wrong reason for the crash but at least it tried - bless its little cybernetic socks. I suppose I have to qualify the statement I made in the the last issue when I said that it does not do anything at all. It does try to help you but it does not report an error if it cannot connect. I suppose that 50% help is better than nothing and M$ do charge a fortune for their telephone assistance.

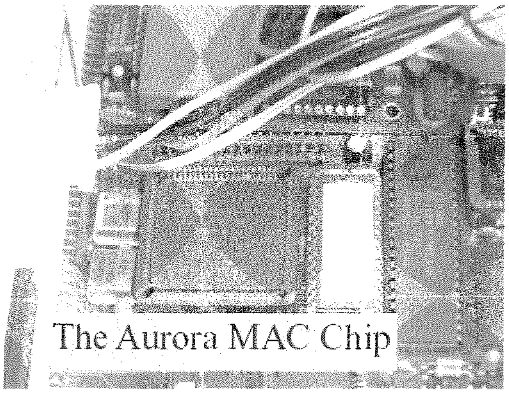
Aurora Scan

My advice to those people who have had problems with the display on their Aurora boards was always to remove the Aurora Scan chip and clean the legs. This always worked on my one although the problems tended to return later. I recently got an email from Mr Bob Spelten, a regular visitor to the Eindhoven shows.

He writes :

"In an earlier e-mail (11.2.03) I asked you about the AURORA-SCAN chip which gave me trouble (flickering, distorted display or no display at all) in spite of frequently cleaning the legs. Now after a few months of using my MinisQL I checked the 'trouble' machine again and found the problem was not the SCAN chip (this time) but the AURMAC chip which had green oxidation on some of its legs!

After cleaning I have not had a screen flicker for days now. This does not change the fact that the AURORA-SCAN chip has one of its legs being held in place only by some conducting paint. So, I still need a replacement because it may not survive the next cleaning."



The Aurora MAC Chip

A new Aurora Scan chip is on its way to him.

I thought that I should pass on this interesting piece of information to other readers who may have similar problems. This chip is, in fact much harder to remove than the other ones because it is not the standard EPROM type arrangement with legs which plug into a socket. It is, in fact one of the small square ones with legs along its sides. To extract this chip you will need to have a special removal tool or, as in the case of the early superHermes replacements, a bent paper clip as supplied by Tony Firshman when he sent out the updates. The chip does have to be removed by hooking the tool (or paper clip) into the two opposite slots and pulling upwards (remember to mark the orientation of the chip so you can replace it). **DO NOT TRY TO LEVER THE CHIP OUT FROM ONE END!** this could cause a crack in the socket and would almost certainly be the end of the board.

The question that remains to be answered is why do these legs corrode in the first place. My MinisQl has resided in the Q Branch Head Office for some time. It was first replaced by the Q 40 when that became a viable alternative and then later with a PC running QPC2. The office is not particularly damp so what is the mechanism by which these chip get into the state they are in? One explanation, proffered by Keith Mitchell, was that the sockets were of the cheaper variety and there was some electrical potential between the two metals. Maybe someone else has an explanation. (Sits back and waits for a long, detailed explantion from Nasta) I would be happy to pass it on. I would also be interested in a permanent cure.

Minnie Cuts Loose

Lau Reeves has just released the source code to the last version of Minerva. This has been a long awaited occurrence and can be obtained from his website.

While this code may be of interest only to programmers and people with access to

EPROM blowers so they can make their own version of Minerva for standard QLs it does, once again, raise the issue of multiple operating systems.

On a platform as large and widely used as the PC having several operating systems and, within that, several different flavours of the same system is, at worst, a minor irritation but with a closed system like our it can be more divisive.

There is very little doubt that many of Lau's routines were tight, well executed, code. He managed to squeeze a lot into a very small area and the results were far superior to the original QL ROMS. Most of the work on the Minerva was done long before SMSQ/E arrived on the scene and there are no high resolution/high colour versions for the QL (although I believe Mark Swift has developed a higher resolution version for the Q40/60).

The release of a swathe of code like this always prompts the several questions What is to be done with it other than use it in the way it was intended? Who will become the arbiter of any development work done on it? And what is the best way to integrate it into the current QL scene?

Integrate don't Segregate!

In my view the best approach here would be to look at the Minerva code objectively and work out how and if the best aspects of it can become part of SMSQ/E. No single programmer has a monopoly on good ideas and the best, most flexible and most useful pieces of software usually come as a result of a collaboration. If we can take the best parts of Minerva and get them into SMSQ/E with all its support for higher screen sizes and larger colour palletes on supported hardware we will have an operating system that has the best aspects of two original and inventive brains. One of the first upgrades that I ever did to my QL was to add a Minerva chip and my machine was all the better for that. On a QL the difference made to both the performance and facilities by a combination of the Minerva and Hermes chips were undeniable.

I have no doubt that SMSQ/E could benefit from an injection of Minerva and the real shame is that there is no way to reverse the process and add some of the SMSQ/E facilities to a Minerva chip - there just is not the room. Having said that, however, the Aurora was designed to support a much bigger ROM and that has, as far as I know, never been utilised.

I know that Nasta has the parts to build several more Aurora boards and the only thing which has stopped this process has been his situation in the US so maybe we will see some development in that area.

New Look Again

I have been using the beta versions of the software written to use the new colour schemes for a few months now and I must say that they certainly do improve the feel of the system. When I rebuilt the main PC after the hard drive failure I reverted to the old colour schemes for a while and it was clearly a very different experience.

The way that a system presents itself to the user is part of the usability of that system. Going back to the old four colour versions of QD and QSpread, two of the most used QL programs on my system, was decidedly uncomfortable now that I am used the the grey backgrounds of the newer versions.

Once the newer versions of these programs are available to us, which I hope, will be by the time you read this, other writers should really be thinking of adding colour support to their programs. I hope that, when the other programmers see the difference that the extra colours have made to QD etc. they will be inspired to adapt theirs too. That process has, to a great degree, already started.

Not To Be Led Up The Garden Upgrade Path

This will, unfortunately lead to a slightly two tiered market with some users able to use the full range of the colours and others who will have to use the older four-colour schemes. On the PC this lack of complete hardware support for the software in question would lead to users being locked out of further upgrades. This has been the way that the PC industry has driven itself forward by forcing users to invest in ever more powerful hardware mostly, I must confess, to run a later version of the same software maybe a nanosecond faster than it was before and with a few new features which most people will never use anyway.

The information I have been given by Jochen is, however, more reassuring for people with the colour support. He does intend to make his programs run on non-high-colour systems for those who only have older hardware or who want to run it on a second system which would not support the newer colour modes. This does not surprise me since people like JMS have always looked after their customers a lot better than the larger corporations in the PC market.

One of the major hurdles in this is that the Window manager (WMAN) will need some revision in order to cope with the new features. It is not sure how this should be done or, indeed by whom, but we are looking at it. In my view the Pointer Environment files should all now become free and this will enable the libraries to give them away but that is another issue.

So We Have The Pallete..... Now We Need A Toolbox

Having all these colours available to programmers is all very well but we do also need the means and the instructions on how to put them into practical programs. Jochen has already done a lot of the work with MENU_REXT and I believe that, given the right instructions, QPTR users should be able to call up the new colour pallete.

Programmers who write their PE programs using EASYptr have, however, been used to using EASYmenu to generate the basic menus and this will only support the older colour schemes. EASYsprite, the program used by this package to generate icons and sprites also has problems under the new schemes and you can not use the new, high definition, icons in EASYptr programs. This is an area which needs urgent attention if current programs are to be brought into the new schemes. If we cannot get the original author, Albin Hessler, to do some of the work to upgrade the original programs then we should try to find someone to write new versions that will be usable with the EASYptr extensions and code.

There could be a lot of improvements and additions made to the last versions of the software which would make this a far better tool.

The System Soldiers On

Every now and then I get surprised by someone who contacts me out of the blue to ask about the availability of older hardware or to buy QPC. You might think that we would know all of the current users by now and most of the lapsed ones too but there are still people who use the QL in their daily lives to run programs for their business and these people are completely out of contact with the current QL scene. In many ways this is not to unthinkable because most of these people have either written the software themselves or have had it written by a third party. They have no use for their QL other than running these programs and they have no wish to expand or add new functions. They are, in fact, the perfect example of kind of user that some members of the user group think that we should be writing software for.

I mentioned before that Wolfgang Urlich writes small applications which run on the PC under QPC2. The user just clicks on the appropriate icon on the PC screen and QPC starts with a boot file which immediately loads the appropriate extensions and runs the application in a Window. To all intents and purposes it looks like a PC program. This is an idea which I have used since to run my VAT Calculator at work.

I know of at least five companies who still use QL based programs to run essential tasks and all of these have bought QPC2 because the rest of the office software is Windoze based. Even with the availability of all the things that current systems have to offer they cannot replace that hand crafted code.

Suit You Sir

One such user contacted me last month and said that he had a suite of programs written by a friend which he ran on his two QLs. They were used to keep track of the stock in his tailoring business. He felt that his two QLs were getting a bit long in the tooth and was worried that they might be failing so he looked us up on the web and found out about QPC2. His idea was to use an old PC in the business and to run the programs under QPC2. He could do the same thing on his PC at home and run the whole process in the same way that he had done on his two QLs.

A swift email with the demo program showed him how easy it was to get QPC2 running but we then had the problem that the systems he was used to had two disk drives with the data on one and the program on the other. I got him to send me a copy of the program and a sample data disk and looked into his problem. The main source of the problem was easily overcome by loading the basic into QD and doing a 'search and replace' substituting 'flp2_' for 'flp1_'.

There are a couple of other errors in the programming which will need a bit of ironing out, however. The usual one is a missing 'END IF' statement.

Old versions of QDOS were not as fussy as SMSQ/E about getting these statement sorted out and you often find that you have to go through the code and find matching pairs. This is not so bad with small pieces of code but this one runs to several thousand lines so it may be a lot harder to sort out.

A 'Honourable Mentions In Despatches' Special

Back Breaking News....

One of the most exhausting things I have had to do at the Hove Workshop this year was bending down and picking up the jaws of people to whom I was demonstrating the new colour schemes and icons. No one thought that a QL screen could look as good as this and the end result is completely stunning. By the time I set up the Q Branch Stall at the show I had installed all of the new goodies that Jochen and Bernd had brought over with them. The really good news for most people was that the upgrades were mostly free with only QSpread and QD incurring a fee.

It is hard to work out who should actually be holding the trophy aloft for this. Certainly Marcel put in a vast amount of work and thought designing the new window manager and the 8 bit pallete. He also burnt the midnight oil for some time to get the Aurora colour driver set up.

Phoebus Dokos did a lot of work too and made most (if not all - it is hard to disentangle who did what) of the stunning 3D icons. Jochen and Bernd also put in a lot of work to make the new versions of QD and QSpread with Bernd also producing a new version of DATAdesign.

There may also be even more people who contributed to the final look and feel of the new interface but it does put QL software into a completely new light.

I will compromise here and, if I have missed anyone's name of the above praise rosta, let me know and I will correct it in the next issue. The Honourable Mention goes to the 'New Interface Design Team'.

Astounding Facts

A recent flyer sent through to the office announced a new colour laser printer to the world. As I glanced through this I noticed something which set off a train of thought and led to this list. The distance between computers as they were when the QL was first launched and today is vast. This list draws no particular conclusion - just enjoy the comparisons and think back to your early days typing in BASIC listings from QL World. Most of the original prices quoted here are from the 1985 QL World magazines.

The new OKI Colour laser printer has a 200MHz processor on board - that is more than twice as fast as the one on a Q80 the fastest QL hardware.

The QL was sold with 128k of RAM most modern laser printers come with at least 4Mb of onboard RAM and can be expanded to much more.

256Kb of very slow RAM on an expansion card for the QL cost £198 [EUR 280,-] - 512Mb of the fastest DDR would set you back £51 [EUR 72,-] today.

A 14" 512 x 256 colour monitor would have cost £299 [EUR 420,-] in 1985 - today you can get 15" 1024 x 768 TFT screen for around £210 [EUR 295,-]

3.5" DD disk drives were £85 [EUR 119,-] (incl. Power supplies - these days the bare drives retail for around £10 [EUR 14,-] and suitable power supplies can be bought at Maplins for less than £5 [EUR 7,-]. The metalwork may be harder to source though.

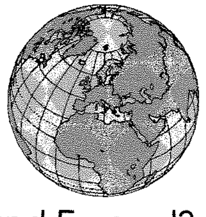
The Mannesmann Tally MT80 9pin dot matrix printer would have set you back £199 [EUR 280,-] in 1985 - today you can get a photo quality inkjet for about £50 [EUR 70,-] or an entry level mon laser printer for £100 [EUR 140,-]. Connecting either to your QL would, however, be a major headache.

And then we have modems....

That stalwart of 1985, the Tandata stack would have cost you £129 [EUR 181,-] in those days and connected you to the wide world of BBS and Prestel at a whopping 9600 baud. Todays modems cost around £20 [EUR 28,-] for an external unit (less than £10 [EUR 14,-] if you want a PCI version for your PC) and allow data transfer at 56K baud if your wires and Internet suppliers are up to it.

As a final word you may be aware that most people more computing power sitting on their desktops today than was used to put a man on the Moon! I leave you to work out what they do with it.

The QL Show Details



Isn't it amazing? Lots of shows coming up this autumn this year all around Europe!?

As I can't squeeze all the show-related information on the next page, here some more details:

Especially for Berchtesgaden, I suggest you go back one and two volumes of QL Today, where Friedemann Oertel (the show organizer) made the effort to give very detailed ideas about what else to do while in the great area of Berchtesgaden!

More details regarding Berchtesgaden for this year's event:
The show will take place in "Hotel Schwabenwirt" (same venue as 2001, opposite to the main station in the centre of Berchtesgaden. Last year's venue was "Tauernhof").
Address info: Hotel Schwabenwirt, Königsseer Str. 1, 83471 Berchtesgaden, GERMANY
Tel. +49-8652-2022, Fax +49-8652-1706

A room has been reserved from Friday evening until Sunday morning, so computers can be brought, built up and locked at night in this room (again, like two years ago).

Davide Santachiara, who is organising the Italian QL Show again, informed us:
This week I asked the permission for the meeting room at the usual place and, even if I will have the official OK only next month, it's fundamentally 99% confirmed.
The place is:
Sala Congressi Circoscrizione, 2 Via Fratelli Cervi, 70 Pieve Modolena, Reggio Emilia, ITALY
You can find the map at
http://www.geocities.com/dsantachiara/mappameetingeng.htm
some round-abouts have been added instead of cross-roads but the way is the same.
I really hope to see you again next October.

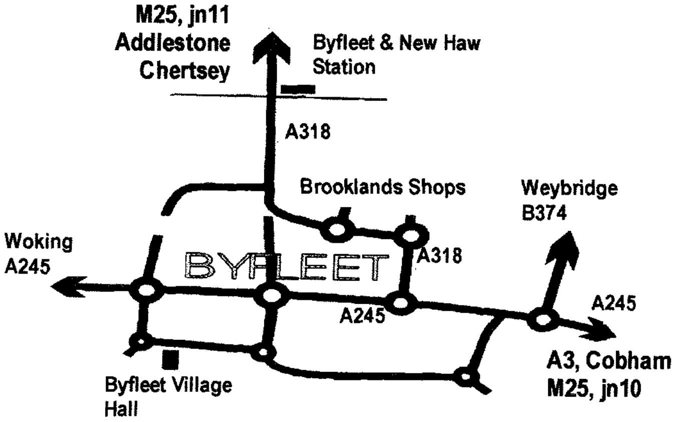
We believe Davide is still looking for good and reasonably priced accomodation and will let us know so that we can publish it in the next issue of QL Today.

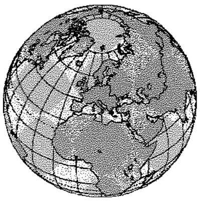
Ken Bain, organiser of the Quanta workshop at Byfleet, has given us the following info in the past:
The Byfleet Village Hall is just inside M25, between jns 10 & 11, just South of A245.
From jn11 go towards Weybridge (East) then turn right (South) onto A318 towards Brooklands, then through the old track onto A245. Right, then left at second roundabout. Left at little roundabout and Hall is on right.

From jn 10, A3 towards London, left onto A245 at next junction, towards Woking. After A318 joins, turn left at second roundabout - see above.

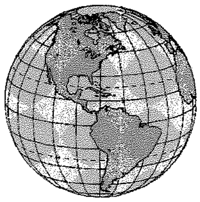
Free Parking, and all the usual attractions. If that isn't enough, Brooklands Museum (aircraft and motor racing) is just up the B374: or there's a Bus Collection on the A245 to Cobham: or I'll explain how to drive to the RHS Gardens at Wisley.
The station is on the Waterloo - Woking line.

J-M-S, Bernd Reinhardt and Marcel Kilgus are already looking into Ferry tickets, so expect us to be there - see you!





The QL Show Agenda



QL Meeting - (NL) Eindhoven

Saturday, 14th of June, 10:00 to 16:00
Pleincollege St. Joris, Roostenlaan 296

QL Meeting - (D) Berchtesgaden

Saturday, 4th of October, 10:00 to 17:00

Hotel Schwabenwirt, Königsseer Str. 1

Same venue as in 2001!

Details on the reverse side!

QL Meeting - (NL) Eindhoven

Saturday, 11th of June, 10:00 to 16:00

Pleincollege St. Joris, Roostenlaan 296

QL Meeting - (I) Reggio Emilia

Sunday, 26th of October, 10:00 to 16:00

Sala Congressi Circoscrizione 2 – Via Fratelli Cervi 97

Same venue as the years before!

Details on the reverse side!

QL Meeting - (GB) Byfleet

Sunday, 9th of November, 10:00 to 16:00

Byfleet Village Hall

Same venue as all the years before!

Details on the reverse side!